

Mateus Soares Marques

**Relatório de Estágio Supervisionado
Laboratório de Sistemas Embarcados e
Computação Pervasiva (Embedded)**

Campina Grande, Paraíba

Junho de 2024

Mateus Soares Marques

Relatório de Estágio Supervisionado
Laboratório de Sistemas Embarcados e Computação
Pervasiva (Embedded)

Relatório de Estágio Supervisionado submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharelado em Ciências no Domínio da Engenharia Elétrica.

Universidade Federal de Campina Grande – UFCG

Centro de Engenharia Elétrica e Informática – CEEI

Departamento de Engenharia Elétrica – DEE

Orientador: Antonio Marcus Nogueira Lima, D. Sc

Campina Grande, Paraíba

Junho de 2024

Mateus Soares Marques

Relatório de Estágio Supervisionado
Laboratório de Sistemas Embarcados e Computação
Pervasiva (Embedded)

Relatório de Estágio Supervisionado submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharelado em Ciências no Domínio da Engenharia Elétrica.

Trabalho aprovado. Campina Grande, Paraíba, 07 de Junho de 2024:

Antonio Marcus Nogueira Lima, D. Sc
Orientador

Jaidilson Jó da Silva, D. Sc.
Convidado

Campina Grande, Paraíba
Junho de 2024

Aos meus pais e minha namorada.

Agradecimentos

Agradeço os meus colegas de projeto, Marina, Breno e Lucas, pelas ideias e apoio no desenvolvimento desse trabalho.

Agradeço o professor Antônio Marcus, pela orientação concedida.

*“O que eu não posso criar, não entendo.”
(Richard Feynman)*

Resumo

No presente relatório estão descritas as principais atividades desempenhadas pelo aluno Mateus Soares Marques, do curso de graduação em Engenharia Elétrica, no estágio realizado no Laboratório de Sistemas Embarcados e Computação Pervasiva(Embedded), durante o período de 06 de fevereiro de 2024 até 20 de maio de 2024. No decorrer deste, serão explanados os fundamentos da tecnologia da caracterização baterias, a elaboração de uma plataforma de caracterização e ciclagem de baterias de Lítio-Polímero e o projeto de uma câmara térmica para realização dos ensaios em um ambiente com temperatura controlada.

Palavras-chave: Baterias, Li-Po, espectroscopia de impedância eletroquímica, câmara térmica.

Abstract

In the present report, the main activities carried out by the student Mateus Soares Marques, an undergraduate in Electrical Engineering, are described. These activities took place during his internship at the Embedded Systems and Pervasive Computing Laboratory, from February 6, 2024, to May 20, 2024. This report will discuss the fundamentals of battery characterization technology, the development of a characterization and cycling platform for Lithium-Polymer batteries, and the design of a thermal chamber for conducting tests in a controlled temperature environment.

Keywords: Batteries, Li-Po, electrochemical impedance spectroscopy, thermal chamber.

Lista de ilustrações

Figura 1 – Foto da fachada do prédio do Laboratório Embedded.	2
Figura 2 – Foto da sala 01 de projetos do prédio do Laboratório Embedded. . . .	3
Figura 3 – Representação do processo de difusão.	5
Figura 4 – Circuito equivalente de Randel	6
Figura 5 – Diagrama de Nyquist típico de uma bateria de LiPo ou Li-íon.	6
Figura 6 – Detalhe da faixa de transferência de carga.	7
Figura 7 – Detalhe das de transferência de carga e difusão.	7
Figura 8 – Dados de espectroscopia típicos na forma de um diagrama de Nyquist. .	9
Figura 9 – EVAL-AD5941BATZ	10
Figura 10 – Keysight EL34243A Bench DC Load.	11
Figura 11 – Diagrama simplificado do circuito do EVAL-AD5941BATZ	12
Figura 12 – Transistores em uma configuração de Darlington.	14
Figura 13 – Diagrama da medição de impedância eletroquímica.	15
Figura 14 – Pasta que contém a API do SensorPal e exemplos de uso.	17
Figura 15 – Detalhes dos equipamentos da série EL30000, da Keysight.	23
Figura 16 – Arquitetura da plataforma de caracterização de baterias.	33
Figura 17 – Placa de relés com 8 canais.	34
Figura 18 – Conexão dos equipamentos à placa de relés.	35
Figura 19 – Hub USB utilizado.	35
Figura 20 – ESP32-Devkit V1.	36
Figura 21 – Esquemático do <i>shield</i> para a ESP.	36
Figura 22 – Desenho da PCB do <i>shield</i> da ESP.	37
Figura 23 – Vista inferior da PCB do <i>shield</i> da ESP.	37
Figura 24 – Vista superior da PCB do <i>shield</i> da ESP.	38
Figura 25 – Plataforma de caracterização de baterias.	43
Figura 26 – Modelo da câmara térmica no Fusion 360.	44
Figura 27 – Vista superior em corte da câmara térmica no Fusion 360.	45
Figura 28 – Vista frontal em corte da câmara térmica no Fusion 360.	45
Figura 29 – Modelo da câmara térmica no Simulink.	46
Figura 30 – Modelo do atuador da câmara térmica no Simulink.	46
Figura 31 – Resfriamento simulado da câmara térmica.	47
Figura 32 – Aquecimento simulado da câmara térmica no Simulink.	47
Figura 33 – Diagrama de Nyquist do teste de espectroscopia.	50
Figura 34 – Magnitude e fase do teste de espectroscopia.	50
Figura 35 – Dados da carga DC coletados no teste.	51

Lista de abreviaturas e siglas

UFCG	Universidade Federal de Campina Grande
CEEI	Centro de Engenharia Elétrica e Informática
Embedded	Sistemas Embarcados e Computação Pervasiva
CATI	Comitê da Área de Tecnologia de Informação
PaqTcPB	Parque Tecnológico da Paraíba

Sumário

1	INTRODUÇÃO	1
1.1	Objetivos	1
1.1.1	Objetivos Específicos	1
1.2	Ambiente de trabalho	2
1.3	Estrutura do relatório	3
2	CARACTERIZAÇÃO DE BATERIAS	4
2.1	Modelagem matemática de baterias	4
2.1.1	Difusão	4
2.1.2	Circuito equivalente	5
2.2	Caracterização	7
2.2.1	Corrente contínua	7
2.2.2	Corrente alternada	8
2.3	Espectroscopia de impedância eletroquímica	8
2.4	Carga eletrônica DC	9
2.5	Câmara térmica	11
3	ESPECTROSCOPIA DE IMPEDÂNCIA	12
3.1	Princípio de funcionamento	12
3.1.1	Sinal de excitação	13
3.1.2	Medindo tensão	14
3.1.3	Calculando a impedância da bateria	14
3.2	Características do EVAL-AD5941BATZ	14
3.3	Programação e captura dos dados	15
3.3.1	API do SensorPal	16
4	CARGA ELETRÔNICA DC	22
4.1	Características da EL34243A	22
4.2	Programação e captura dos dados	23
4.2.1	Linguagem SCPI	24
4.2.1.1	Formato dos comandos	25
4.2.1.2	Separadores de comando	26
4.2.1.3	Terminadores de comando	27
4.2.1.4	Comandos comuns ao IEEE-488.2	27
4.2.1.5	Tipos de Parâmetros SCPI	27
4.2.1.6	Parâmetros de Canal	27

4.2.1.7	Parâmetros Numéricos	28
4.2.1.8	Parâmetros Discretos	28
4.2.1.9	Parâmetros Booleanos	28
4.2.2	Comandos de interesse	28
5	COMPOSIÇÃO DO SISTEMA	32
5.1	Arquitetura do Sistema	32
5.2	Conexão entre os equipamentos	32
5.2.1	Conexão elétrica	32
5.2.2	Conexão de dados	32
5.3	Controle da placa de relés	34
5.3.1	Firmware da ESP32-Devkit V1 e software de comunicação	39
5.4	Montagem	42
6	CÂMARA TÉRMICA	44
6.1	Modelagem 3D	44
7	RESULTADOS	49
8	CONSIDERAÇÕES FINAIS	52
	REFERÊNCIAS	53

1 Introdução

O presente relatório é o documento de descrição das principais atividades realizadas pelo estudante de graduação em Engenharia Elétrica, Mateus Soares Marques, durante o Estágio Supervisionado no Laboratório de Sistemas Embarcados e Computação Pervasiva (Embedded). O estágio teve carga horária de 12 horas semanais e ocorreu no período de 06 de fevereiro a 20 de maio de 2024, totalizando 450 horas. O estudante foi orientado pelo professor Antonio Marcus Nogueira Lima e supervisionado pelo professor Jaidilson Jó da Silva.

A elaboração, apresentação e avaliação da disciplina Estágio Supervisionado ou Integrado são obrigatórias para os alunos concluintes, conforme previsto no Projeto Pedagógico do Curso de Engenharia Elétrica do Centro de Engenharia Elétrica e Informática (CEEI) da Universidade Federal de Campina Grande (UFCG). O objetivo é proporcionar ao estudante uma experiência profissional na qual seja possível conhecer e executar atividades associadas à Engenharia Elétrica, consolidando, assim, os conhecimentos adquiridos no decorrer do curso.

Com realização de forma presencial, as principais atividades desenvolvidas referem-se ao projeto e implementação de uma plataforma de caracterização e ciclagem de baterias de LiPo Li-íon, por meio de descargas de corrente contínua e espectroscopia de impedância.

1.1 Objetivos

O objetivo principal definido para o estágio supervisionado, realizado no Laboratório de Embedded, foi a configuração, desenvolvimento de uma plataforma de ciclagem e caracterização de baterias de LiPo e LI-íon, composto por um conjunto de *hardware* e *software*, baseado em ciclagem e espectroscopia de impedância eletroquímica.

1.1.1 Objetivos Específicos

Os objetivos específicos definidos foram:

1. Estudar a estrutura e técnicas de caracterização de baterias de LiPo e Li-íon;
2. Realizar o estudo e programação de um equipamento de espectroscopia de impedância eletroquímica;
3. Realizar o estudo e programação de uma carga eletrônica DC;

4. Projetar o sistema de chaveamento entre o equipamento de espectroscopia, a carga eletrônica e o carregador de baterias;
5. Projetar a câmara térmica para aclimação das baterias em teste.

1.2 Ambiente de trabalho

O Laboratório de Sistemas Embarcados e Computação Pervasiva (Embedded) faz parte do CEEI/UFCG, em Campina Grande, Paraíba e foi fundado em dezembro de 2005, o laboratório ocupa um prédio de 600 metros quadrados no campus da UFCG. Na Figura 1 é apresentada uma fotografia da fachada do prédio.

Figura 1 – Foto da fachada do prédio do Laboratório Embedded.



Fonte: Retirado de embedded.ufcg.edu.br.

Através da UFCG, o Laboratório Embedded é credenciado no Comitê da Área de Tecnologia de Informação (CATI) para receber recursos da Lei de Informática, tendo o Parque Tecnológico da Paraíba (PaqTcPB) como interveniente financeiro também credenciado no CATI.

A equipe do Laboratório Embedded é formada por professores e pesquisadores do CEEI da UFCG. O laboratório conta com cerca de 60 colaboradores, incluindo alunos de doutorado, mestrado e graduação, dos cursos de Engenharia Elétrica e Ciência da Computação.

A missão do Embedded é avançar no estado da arte nas áreas de sistemas embarcados e computação pervasiva, promovendo ações que permitam que tais avanços tragam benefícios para a sociedade através de parcerias com grandes empresas. Para isso, tem-se uma equipe formada por pesquisadores focados na produção de conhecimento e na

aplicação deste conhecimento na resolução de problemas reais da indústria, equilibrando perspectivas acadêmicas com as necessidades de mercado.

O foco está na concepção, implementação e experimentação de soluções para sistemas embarcados e computação pervasiva, incluindo métodos, técnicas, arcabouços, ferramentas e aplicações, com o objetivo de proporcionar tecnologias invisíveis para o usuário, acessíveis a todo momento, em todo lugar e enquanto o usuário se movimentar.

As atividades foram desenvolvidas utilizando como ambiente principal a Sala 01, uma das salas de projetos - P & D *Hardware*, do Embedded. Na Figura 2 é apresentada uma foto da sala citada, apresentando o ambiente de trabalho e os equipamentos disponíveis.

Figura 2 – Foto da sala 01 de projetos do prédio do Laboratório Embedded.



Fonte: Autoria própria.

1.3 Estrutura do relatório

O presente relatório está estruturado em 8 capítulos, incluindo esta introdução, conforme será descrito a seguir. Neste capítulo, é apresentada uma breve introdução, os objetivos do estágio e o local de trabalho. No capítulo 2, discute-se a estrutura e técnicas de caracterização de baterias de LiPo e Li-íon. No capítulo 3, é apresentado o equipamento de espectroscopia de impedância eletroquímica EVAL-AD5941BATZ ([Analog Devices, 2022](#)), suas características e modo de uso. No capítulo 4, é apresentada a carga eletrônica DC EL34243A ([Keysight Technologies, 2024c](#)), suas características e modo de uso. No capítulo 5 é apresentado o projeto da plataforma automática de caracterização e ciclagem de baterias de Li-íon, usando os equipamentos abordados nos capítulos 3 e 4. No capítulo 6 discute-se a modelagem e o projeto de uma câmara térmica usando tecnologia Peltier. No capítulo os resultados obtidos do projeto da plataforma de caracterização são apresentados e discutidos. Por fim, no capítulo 8 conclui-se este documento resumizando as atividades desempenhadas no estágio e os próximos passos do projeto.

2 Caracterização de baterias

As baterias de Lítio-Polímero (Li-Po) são uma subcategoria das baterias de lítio-ion, diferenciando-se principalmente pela utilização de um eletrólito polimérico sólido em vez de líquido. Este eletrólito sólido permite que as baterias Li-Po tenham uma forma mais flexível e menos suscetível a vazamentos, comparativamente às suas contrapartes de lítio-ion com eletrólito líquido ([Clean Energy Institute, 2024](#)).

Estruturalmente, as baterias Li-Po são compostas por camadas de eletrodos e eletrólito empilhadas ou enroladas. O ânodo é tipicamente feito de grafite, enquanto o cátodo é composto por um material à base de metal de lítio, como fosfato de ferro-lítio ou óxido de cobalto-lítio. A característica distintiva da bateria Li-Po é a substituição do separador poroso impregnado de eletrólitos líquidos por um polímero sólido que conduz íons de lítio entre os eletrodos ([Power Sonic, 2024](#)).

As baterias Li-Po são valorizadas por sua alta densidade de energia e capacidade de operar em uma ampla faixa de temperaturas. Elas também podem ser projetadas para oferecer altas taxas de descarga, tornando-as adequadas para aplicações que requerem uma liberação rápida de energia, como em aeronaves de controle remoto e veículos elétricos.

2.1 Modelagem matemática de baterias

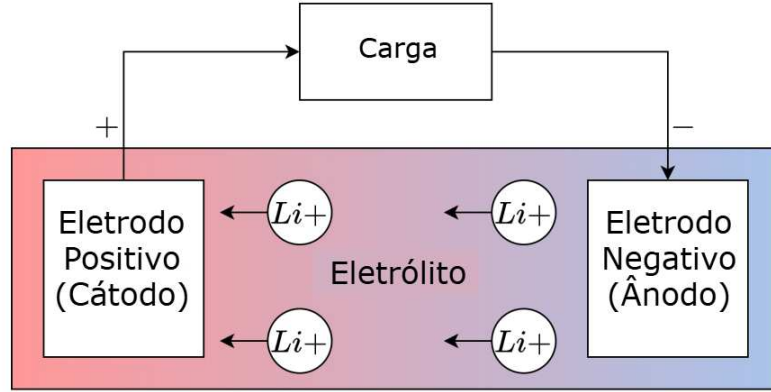
Os modelos matemáticos considerados neste trabalho são o modelo de Rakhmatov-Vrudhula, utilizado para modelar a difusão de íons de lítio, fornecendo uma solução para estimativa do estado de carga da bateria (SOC), e o modelo de circuito equivalente, que fornece dados acerca da impedância da bateria para as várias componentes de corrente em frequência que podem passar pelas células durante o uso.

2.1.1 Difusão

Uma bateria, ou acumulador eletroquímico, é definida como um dispositivo capaz de transformar energia química em energia elétrica. A estrutura típica de uma bateria consiste na associação em série ou paralelo de uma ou mais células eletroquímicas. Cada célula é composta por um eletrodo positivo, conhecido como cátodo, e um eletrodo negativo, chamado ânodo, separados por um eletrólito. A energia química é convertida em energia elétrica através das reações químicas que ocorrem entre os eletrodos e os íons no eletrólito, também referidos como espécies eletroativas.

Durante o processo de descarga nas células de bateria baseadas em lítio, ocorre a difusão de íons de lítio do ânodo para o cátodo, como ilustrado na figura 3. Na fase de

Figura 3 – Representação do processo de difusão.



Fonte: (TAVARES, 2023).

recarga, a difusão ocorre em sentido oposto, isto é, do cátodo para o ânodo, e é bidirecional, variando conforme o regime de operação da bateria.

O modelo de difusão Rakhmatov-Vrudhula tem sido utilizado para modelar a dinâmica de difusão em baterias de íon-lítio na literatura, oferecendo uma solução precisa para o problema de estimativa do estado de carga com um baixo custo computacional. O modelo resultante resolve um conjunto de equações diferenciais que descrevem o processo de difusão dentro da bateria. Os parâmetros do modelo dependem exclusivamente do conhecimento da capacidade total da bateria e do coeficiente de difusão.

Embora existam diferenças nos materiais utilizados na construção de baterias de Li-ion e LiPo, o fenômeno de difusão é descrito pelas leis de Fick em ambos os casos, o que sugere que a difusão opera de maneira semelhante para ambas as baterias. As equações que descrevem a difusão segundo esse modelo são (RAKHMATOV; VRUDHULA, 2001)

$$H(s) = \frac{\sigma(t)}{I_b(s)} = \frac{1}{s} + 2 \sum_{k=0}^{\infty} \frac{1}{s + \beta^2 k^2} \quad (2.1)$$

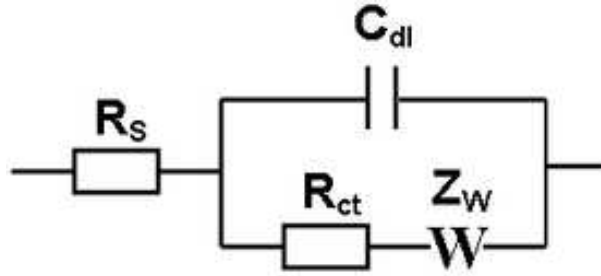
$$A(t) = 100 \left(\frac{\alpha - \sigma(t)}{\alpha} \right) \% \quad (2.2)$$

onde $H(s)$ é a função de transferência da corrente I para a carga perdida σ e $A(t)$ é o estado de carga (SOC) da bateria. α e β são parâmetros do modelo que podem ser estimados uma vez que providos os dados dos ensaios.

2.1.2 Circuito equivalente

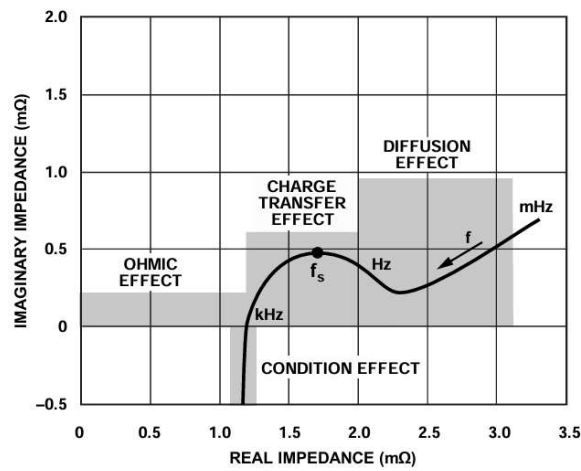
Os modelos de circuito equivalente representam o comportamento elétrico global da bateria. Existem diversas representações do comportamento elétrico sob forma de diferentes arranjos de elementos passivos. Uma das representações mais simples é o circuito de Randel, ilustrado na figura 4

Figura 4 – Circuito equivalente de Randel



Fonte: (KYUNGSOON, 2022).

Figura 5 – Diagrama de Nyquist típico de uma bateria de LiPo ou Li-íon.

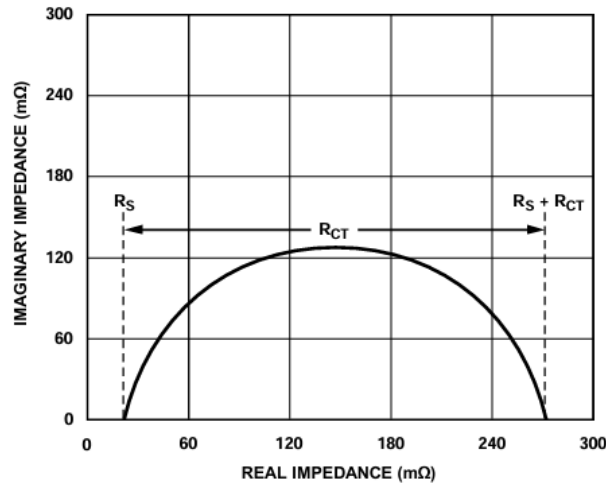


Fonte: (Analog Devices, 2022).

A resistência em série (R_s) representa a resistência total do sistema, incluindo a resistência do eletrólito, dos contatos e dos eletrodos. A capacitância da dupla camada elétrica (CPE), modela a capacitância formada na interface entre o eletrodo e o eletrólito. Essa capacitância reflete a capacidade de armazenamento de carga nessa interface e está relacionada à reação de adsorção/dessorção de íons na superfície do eletrodo. Normalmente, em paralelo com a resistência de transferência de carga (R_{ct}), que representa a resistência associada ao processo de transferência de carga na interface eletrodo/eletrólito. Essa resistência está relacionada à taxa de reação eletroquímica na interface e afeta a eficiência eletroquímica da célula. Em alguns exemplo, também é representado a impedância do elemento Warburg (Z_w), também conhecido como difusor, modela o processo de difusão de íons ou moléculas no eletrólito.

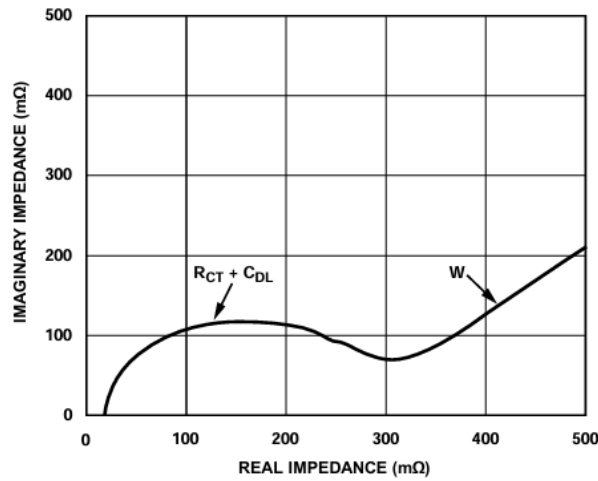
Os três efeitos capturados por esses componentes podem ser visualizados num diagrama de Nyquist, como mostrado nas figuras 5 à 7.

Figura 6 – Detalhe da faixa de transferência de carga.



Fonte: ([Analog Devices, 2022](#)).

Figura 7 – Detalhe das de transferência de carga e difusão.



Fonte: ([Analog Devices, 2022](#)).

2.2 Caracterização

Para identificar parâmetros em modelos de bateria, utilizam-se métodos baseados em corrente contínua (DC) e corrente alternada (AC). Essas abordagens permitem extrair parâmetros críticos para análise de baterias, como estado de carga (SOC), estado de saúde (SOH) e resistência interna, a partir de medições de tensão, corrente, temperatura e impedância.

2.2.1 Corrente contínua

Em testes de DC, os cicladores executam ciclos de carga e descarga regulados para avaliar desgaste, capacidade e longevidade das baterias. Esses testes são realizados de maneiras que facilitam a identificação de parâmetros do modelo sem depender dos testes

de AC. Recomenda-se realizar esses testes em ambientes controlados termicamente para estudar os efeitos da temperatura no desempenho e segurança das baterias, especialmente quando se opera acima das correntes especificadas tecnicamente.

A tensão de circuito aberto (VOC) é determinada descarregando a bateria com uma corrente DC de baixa amplitude até a tensão limite especificada, de modo que a queda de tensão devido à resistência interna seja mínima. Alternativamente, realiza-se uma descarga contínua que simula o perfil de uso seguido por um período de descanso. Contudo, o método de corrente pulsada pode acelerar o desgaste da bateria devido ao uso de correntes que excedem sua capacidade nominal, mesmo que brevemente, sendo esta a principal limitação deste método.

2.2.2 Corrente alternada

Nos testes de AC, a espectroscopia de impedância eletroquímica (EIS) é utilizada para analisar características elétricas da bateria em várias frequências, fornecendo informações sobre resistências e capacitâncias internas relacionadas à estrutura e processos de difusão. O teste utiliza corrente alternada de baixa amplitude para manter o sinal dentro de uma faixa linear durante a varredura de frequências. A EIS é essencial para compreender o comportamento dinâmico das baterias, contribuindo para a caracterização precisa e para o desenvolvimento de estratégias de gerenciamento de energia, melhorando a eficiência e segurança dos sistemas.

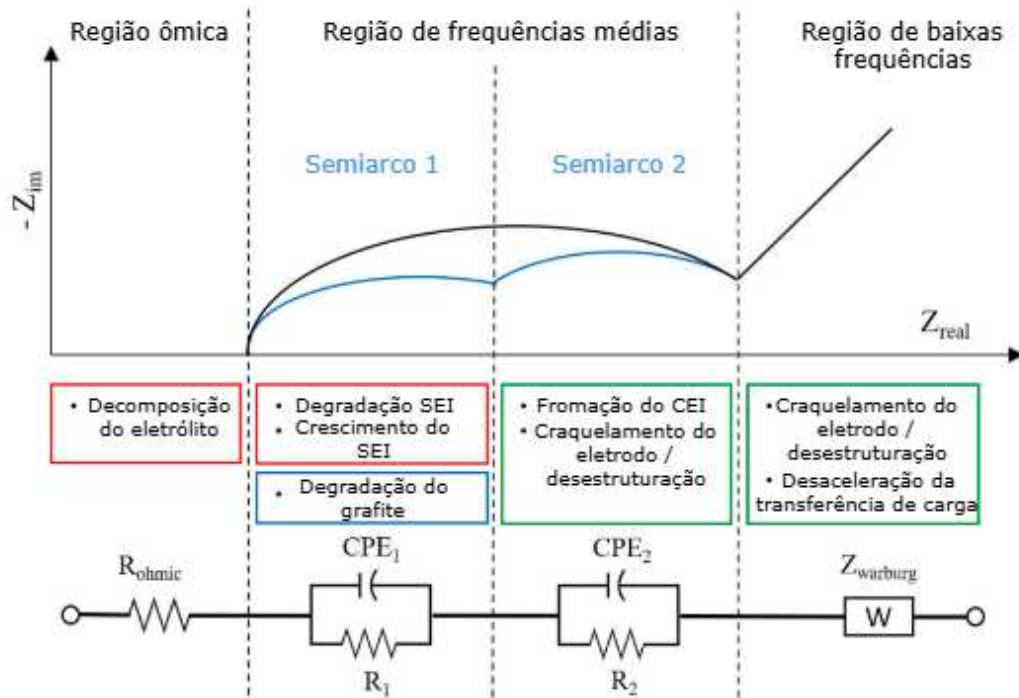
2.3 Espectroscopia de impedância eletroquímica

A espectroscopia de impedância eletroquímica (EIS) proporciona uma análise detalhada das propriedades elétricas da interface eletrodo-solução, inserindo-se nas técnicas de caracterização eletroquímica interfacial. Utilizando métodos associados, é possível investigar o comportamento de um sistema ao avaliar as velocidades de processos interconectados (MEDDINGS et al., 2020).

A EIS, desempenha um papel importante na caracterização de baterias através de modelos híbridos, como o circuito equivalente de Randel combinado com o modelo de difusão de Rakhmatov-Vrudhula. Esta abordagem fornece uma análise abrangente das características eletroquímicas das baterias, possibilitando a determinação de parâmetros importantes para o desempenho da bateria, como a resistência da solução, a resistência de transferência de carga, a capacidade e a eficiência de transporte de íons.

Na região de alta frequência do espectro EIS, a resistência em série (R_s) está relacionada com a perda total de carga do sistema, incluindo a resistência do eletrólito, dos contatos e dos eletrodos. A interface de elétrons e íons em estado sólido (SEI) não é representada diretamente no circuito de Randles, mas é uma camada fina que se forma

Figura 8 – Dados de espectroscopia típicos na forma de um diagrama de Nyquist.



Fonte: (IURILLI; BRIVIO; WOOD, 2021).

na superfície do eletrodo durante as primeiras etapas do ciclo de carga e descarga da célula eletroquímica. A SEI é crucial para a estabilidade operacional da célula, pois previne reações indesejadas entre o eletrólito e o eletrodo e facilita o transporte de íons. Na região intermediária do espectro, a resistência de transferência de carga na interface eletrodo/eletrólito é modelada, usando um capacitor para simular a capacitância da dupla camada elétrica e um resistor para representar a resistência de transferência de carga.

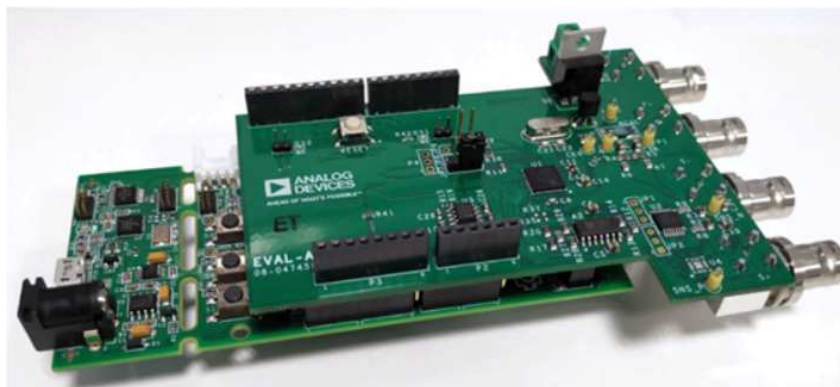
Na região de baixas frequências, a impedância de Warburg, associada à difusão de espécies eletroativas nos eletrodos e na barreira interfacial, é frequentemente utilizada para obter insights sobre a dinâmica de transporte de carga. A modelagem desses fenômenos permite aprimorar o entendimento do comportamento dinâmico das baterias e contribuir para o desenvolvimento de abordagens para otimizar a eficiência e o gerenciamento de energia.

A empresa Analog Devices possui um equipamento de EIS para baterias, o EVAL-AD5941BATZ, que é disponibilizado na forma de *shield* para plataformas compatíveis com Arduino UNO. Esse equipamento foi utilizado nesse trabalho e aparece na figura 9.

2.4 Carga eletrônica DC

Cargas eletrônicas DC programáveis, como o modelo EL34243A da Keysight, são dispositivos usados para testar e simular condições de carga em fontes de energia DC como

Figura 9 – EVAL-AD5941BATZ



Fonte: ([Analog Devices, 2022](#))

baterias, células de combustível, e fontes de alimentação. Estes equipamentos são essenciais para engenheiros e pesquisadores que trabalham no desenvolvimento e verificação de sistemas de energia elétrica, pois permitem a aplicação de cargas controladas e específicas sobre os sistemas em teste.

Uma carga eletrônica DC programável é capaz de imitar o consumo de energia de um componente eletrônico ou sistema completo, oferecendo a flexibilidade de ajustar a corrente, a tensão, e a potência conforme necessário durante o teste. Este tipo de dispositivo normalmente inclui funcionalidades para configuração de diversos modos de operação, como constante corrente (CC), constante tensão (CV), constante resistência (CR), e constante potência (CP), que permitem testar a resposta do dispositivo sob teste (DUT) sob diferentes cenários de carga.

O modelo EL34243A da Keysight, em particular, é projetado para oferecer alta precisão e versatilidade, suportando uma ampla gama de correntes e tensões. Este modelo pode ser utilizado para realizar testes automatizados por meio de interfaces de programação, permitindo integração com sistemas automatizados de teste e validação. Além disso, cargas como esta frequentemente incluem capacidades de medição e registro de dados, possibilitando uma análise detalhada da performance do DUT sob variadas condições de operação. O modelo da Keysight traz recurso de operação do dispositivo via interface serial, o que permite a realização de uma série de descargas programadas.

A carga eletrônica DC permite a realização de diferentes tipos de ensaio para caracterização de baterias, como ensaios de corrente constante, potência constante e corrente pulsada. O modelo EL34243A da Keysight foi utilizado nesse trabalho e aparece na figura 10.

Figura 10 – Keysight EL34243A Bench DC Load.



Fonte: ([Keysight Technologies, 2024c](#))

2.5 Câmara térmica

Uma câmara térmica de temperatura controlada implementada com tecnologia Peltier utiliza módulos termoeletrônicos para o controle preciso da temperatura. Esses módulos funcionam com base no efeito Peltier, onde a corrente elétrica é aplicada através de uma junção de dois materiais diferentes, gerando calor em um lado e frio no outro. A câmara é capaz de atingir e manter temperaturas específicas rapidamente e de forma eficiente, sendo ideal para aplicações que exigem condições de temperatura estável e controlada, como testes de materiais ou dispositivos eletrônicos sob variadas condições térmicas.

No contexto de ciclagem e caracterização de baterias, uma câmara térmica controlada é essencial para simular condições ambientais diversas durante testes de bateria. A câmara permite aplicar ciclos de temperatura que imitam as condições reais de uso, avaliando como as baterias respondem a variações térmicas durante o carregamento e descarregamento. Isso ajuda a determinar a vida útil, capacidade e segurança das baterias sob diferentes regimes de temperatura, garantindo que os resultados dos testes sejam relevantes para aplicações práticas.

Nesse trabalho fez-se um projeto de uma câmara térmica para ser usada juntamente com a plataforma automatizada de caracterização e ciclagem.

3 Espectroscopia de Impedância

O equipamento de espectroscopia estudado nas atividades do estágio foi o EVAL-AD5941BATZ, da Analog Devices.

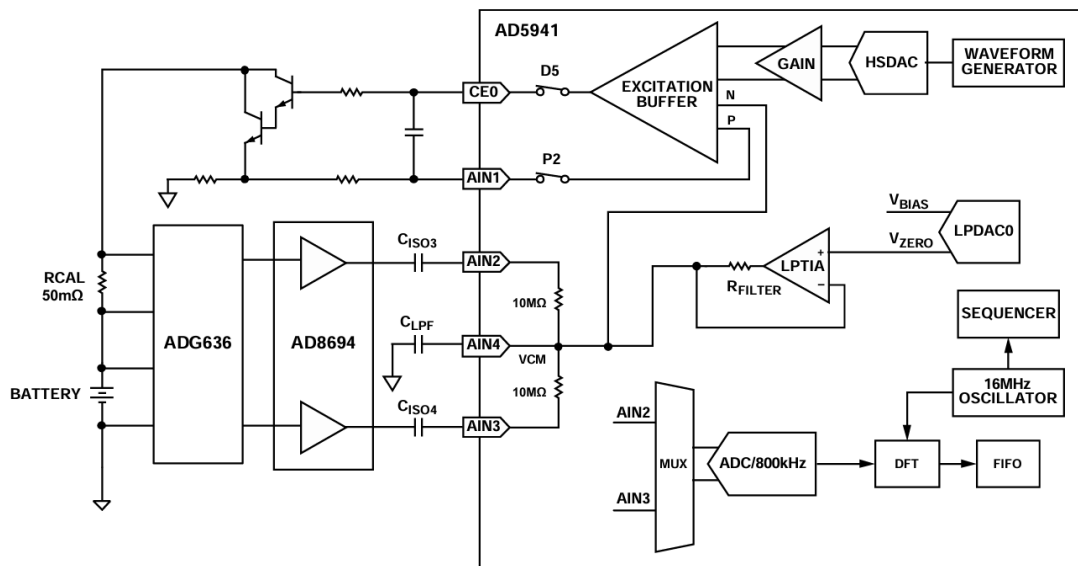
3.1 Princípio de funcionamento

O circuito mostrado na figura 11 é um sistema de medição de espectroscopia de impedância eletroquímica (EIS) para caracterizar baterias de íon de lítio (Li-Ion) e outros tipos implementado pelo EVAL-AD5941BATZ. EIS é uma técnica de perturbação segura usada para examinar processos internos em sistemas eletroquímicos. O sistema mede a impedância de uma célula de bateria em uma faixa de frequências. Os dados podem determinar o estado de saúde (SOH) e estado de carga (SOC) de uma bateria. Usando uma frente de entrada analógica de baixa potência (AFE), o sistema é projetado para excitar e medir corrente, tensão ou resposta de impedância de uma bateria.

O envelhecimento leva à degradação do desempenho e mudanças irreversíveis na química da bateria. A impedância aumenta linearmente com o declínio da capacidade. Monitorar o aumento da impedância de uma bateria usando EIS pode determinar o SOH e se a bateria precisa ser substituída, reduzindo tanto o tempo de inatividade do sistema quanto os custos de manutenção.

Baterias requerem uma excitação de corrente, não de tensão, e os valores de

Figura 11 – Diagrama simplificado do circuito do EVAL-AD5941BATZ



Fonte: (Analog Devices, 2022).

impedância são pequenos, na faixa de mili-ohm. Este sistema inclui o circuito necessário para injetar uma corrente na bateria e permite a calibração e detecção das pequenas impedâncias encontradas em baterias.

A solução de impedância de bateria utilizando AD5941 da Analog Devices é o núcleo do sistema de medição de espectroscopia de impedância eletroquímica (EIS). O AD5941 consiste em um circuito de baixa largura de banda e um de alta largura de banda, um conversor analógico-digital (ADC) de alta precisão e uma matriz de chaves programável.

O circuito de baixa largura de banda contém um conversor digital-analógico (DAC) de baixa potência que gera V_{ZERO} e V_{BIAS} , e um amplificador de transimpedância de baixa potência para converter correntes de entrada em voltagem.

O circuito de alta largura de banda é utilizado para medidas de EIS, com um DAC de alta velocidade que gera sinais de excitação AC para medições de impedância e um amplificador de transimpedância de alta velocidade que converte correntes de alta frequência até 200kHz em voltagens que são medidas pelo ADC.

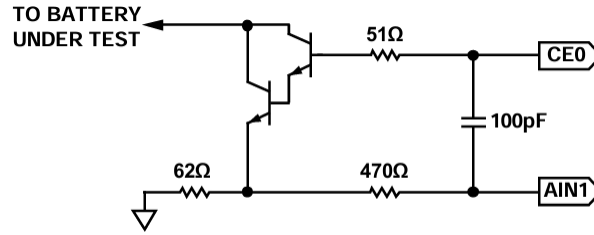
A matriz de chaves é uma série de interruptores programáveis que permitem a conexão de pinos externos ao DAC de alta velocidade e ao amplificador de transimpedância de alta velocidade, fornecendo uma interface para conectar um eletrodo externo. A matriz também adiciona flexibilidade na conexão de eletrodos.

As baterias geralmente possuem impedâncias na faixa de mili-ohms e exigem um resistor de calibração, R_{CAL} , de valor similar. O R_{CAL} de 50m Ω neste circuito é muito pequeno para o AD5941 medir diretamente. Devido ao pequeno R_{CAL} , utiliza-se um estágio de ganho externo com o AD8694 para amplificar o sinal recebido. O AD8694 possui desempenho de ruído ultrabaixo e parâmetros de corrente de viés e fuga baixos, essenciais para aplicações de EIS. Além disso, usar o mesmo amplificador tanto para o R_{CAL} quanto para a bateria real pode ajudar a compensar erros provenientes de cabos, capacitores de acoplamento e do próprio amplificador.

3.1.1 Sinal de excitação

O AD5941 utiliza um gerador de ondas, um DAC de alta velocidade (HSDAC) e um amplificador de excitação para gerar sinais de onda senoidal. A frequência desses sinais é programável de 0,015Hz até 200kHz. O sinal é aplicado à bateria através do pino CEO e uma configuração externa de transistores Darlington, como ilustrado na figura 12. Um amplificador de corrente é necessário porque o buffer de excitação pode fornecer no máximo 3 mA, enquanto uma bateria típica pode requerer até 50 mA.

Figura 12 – Transistores em uma configuração de Darlington.



Fonte: ([Analog Devices, 2022](#)).

3.1.2 Medindo tensão

Existem duas etapas para medir a tensão. Primeiro, a queda de tensão no resistor de calibração R_{CAL} é medida. Em seguida, a tensão através da bateria é medida. A queda de tensão em cada componente é muito pequena, na ordem de microvolts. Os dados obtidos são enviados diretamente dos amplificadores de ganho AD8694 para o ADC no AD5941 nos pinos AIN2 e AIN3. Utilizando um acelerador de hardware para a Transformada Discreta de Fourier (DFT), um DFT é realizado nos dados do ADC, onde os números reais e imaginários são calculados e armazenados na fila FIFO tanto para a tensão do resistor de calibração quanto para a tensão da bateria. O ADG636 multiplexa a bateria e o R_{CAL} para o estágio de ganho AD8694.

As pequenas correntes de injeção de carga e de fuga do interruptor ADG636 são necessárias para anular qualquer capacitância parasita nos pinos de entrada do AD5941. Como os pinos AIN2 e AIN3 são usados para medições tanto do R_{CAL} quanto da bateria, o caminho do sinal para a medição de impedância é ratiométrico. A figura 13 traz um diagrama simplificado do processo de medição da impedância.

3.1.3 Calculando a impedância da bateria

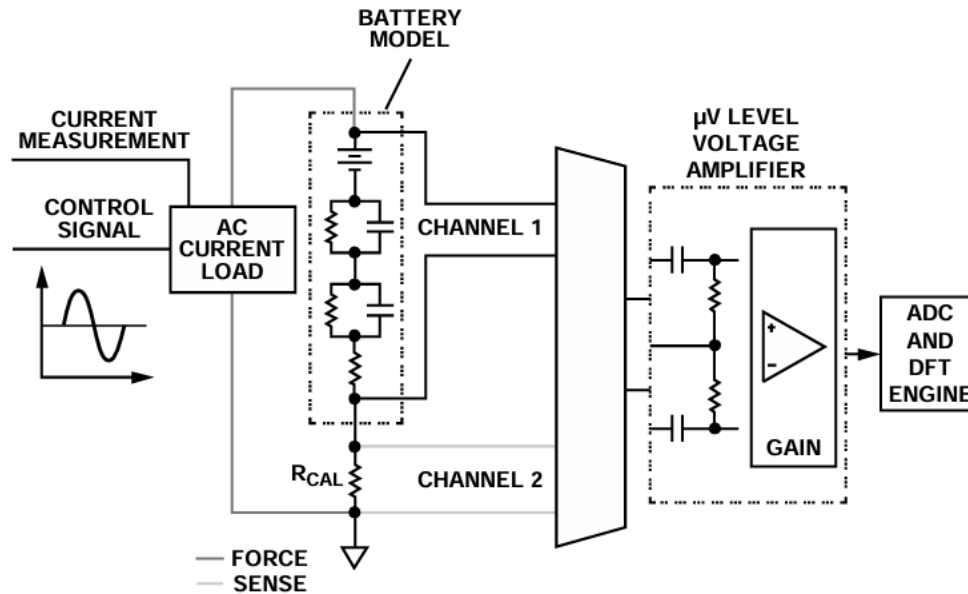
Medições de EIS são ratiométricas. Para medir uma impedância desconhecida $Z_{UNKNOWN}$, um sinal de corrente alternada é aplicado através do resistor conhecido, R_{CAL} , e a tensão de resposta, V_{R_CAL} , é medida. Em seguida, o mesmo sinal é aplicado através da impedância desconhecida, $Z_{UNKNOWN}$, e a tensão de resposta, $V_{UNKNOWN}$, é medida. A impedância desconhecida pode ser calculada usando a seguinte equação:

$$Z_{UNKNOWN} = \left(\frac{V_{UNKNOWN}}{V_{R_CAL}} \right) \times R_{CAL} \quad (3.1)$$

3.2 Características do EVAL-AD5941BATZ

As principais características do equipamento de EIS usado são:

Figura 13 – Diagrama da medição de impedância eletroquímica.



Fonte: ([Analog Devices, 2022](#)).

- Amplitude DC e AC da excitação são configuráveis para valores de 0 a 1200mV_{DC} e de 0 a 600mV_{AC} aplicados ao par Darlington;
- Varredura em frequência configurável em intervalos de 0.1Hz à 200kHz;
- Varreduras em frequências de sub-Hz são possíveis, mas o sistema é sintonizado para varreduras acima de 1Hz. Valores abaixo disso geram dados ruidosos devido ao ruído de $1/f$ produzido pela etapa de amplificação externa no AD8694;
- O *sweep* pode ser realizado em escala linear e logarítmica de frequências com até 2147483647 pontos;
- Se o usuário preferir configurar o próprio firmware para a plataforma micro-controlada onde se deseja colocar o shield EVAL-AD5941BATZ, pode-se configurar a quantidade de amostras usadas na DFT e o tipo de filtragem aplicada aos dados passados à transformada, que são parâmetros do *front-end* analógico AD5941;
- Compatibilidade com plataformas micro-controladas compatíveis com Arduino UNO que operem em 3.3V.

3.3 Programação e captura dos dados

O uso da plataforma EVAL-AD5941BATZ pode ser realizada de diversas formas. A Analog Devices recomenda a integração desse *shield* com a placa EVAL-ADICUP3029, também da Analog Devices. Esse último equipamento também foi adquirido. Com ele é

possível usar todas as soluções propostas pela empresa para o equipamento de EIS. São elas

- Software SensorPal - Interface gráfica para configuração e coleta de dados de placas EVAL;
- API do SensorPal - API em Python para realização programática dos ensaios disponíveis no software SensorPal;
- Firmware escrito em C baseado nos exemplos do repositório da Analog Devices no Github ([Analog Devices Inc., 2024](#)) para plataformas baseadas no chip AD5941. Lá existe um exemplo específico para programação do firmware do EVAL-ADICUP3029 para uso com o shield EVAL-AD5941BATZ.

A solução escolhida nesse projeto foi a da API do SensorPal, pois com ela é possível integrar facilmente o equipamento de EIS ao projeto sem incorrer nas dificuldades associadas à programação do firmware.

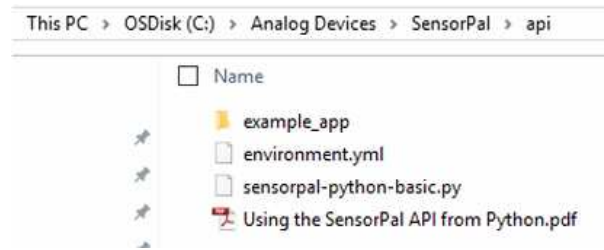
3.3.1 API do SensorPal

A API expõe a camada de software do SensorPal que fica abaixo da interface gráfica. Assim, com a API é possível executar todas as funcionalidades disponíveis na interface programaticamente. Essas funcionalidades são:

- Seleção da técnica de medição;
- Modificar parâmetros da técnica escolhida;
- Selecionar um tipo de plotagem disponível para a técnica;
- Selecionar a porta serial à qual está conectada o equipamento;
- Iniciar e parar um processo de medição. Com essa última funcionalidade é possível capturar os dados produzidos pelo equipamento em tempo real, sendo esses dados formatados de acordo com o tipo de plotagem escolhida.

A API é implementada em C# (.NET) e é linkada ao python através da biblioteca `pythonnet`. É possível encontrar os recursos necessários para instalar e usar a API dentro do diretório de instalação do SensorPal, numa pasta chamada `api`, como mostrado na figura 14. Lá exemplos de uso bem como a documentação da API são disponibilizados. As funções mais importantes da API são listadas abaixo:

Figura 14 – Pasta que contém a API do SensorPal e exemplos de uso.



Fonte: Autoria própria.

- `public void OpenConnection(string portName):`

Esta função tentará abrir a porta COM especificada. Um comando bem-sucedido deve ser acompanhado por uma chamada para `CloseConnection()` para liberar a porta COM.

- `public void CloseConnection():`

Esta função fechará a porta COM que foi aberta por uma chamada bem-sucedida ao `OpenConnection()`.

- `public SequenceSpec GetDefaultTechniqueParameters(string technique_string):`

Esta função recupera o conjunto padrão de parâmetros para uma técnica de medição específica. Os parâmetros podem ser modificados posteriormente por chamadas para `UpdateTechniqueParameter()`, e o conjunto final de parâmetros pode ser passado para `Measure()` para realizar a medição.

Parâmetros:

`technique_string` - uma string indicando a técnica de medição a ser usada.

- `public SequenceSpec UpdateTechniqueParameter(string technique_string, SequenceSpec sequenceSpec, string subtoolName, string value):`

Esta função modificará um parâmetro dentro do conjunto padrão de parâmetros para uma técnica de medição. O conjunto final de parâmetros pode ser usado para executar a medição. Os parâmetros correspondem aos controles apresentados ao usuário na interface gráfica do SensorPal.

Parâmetros:

`technique_string` - uma string indicando a técnica usada;
`sequenceSpec` - uma instância de `SequenceSpec` onde o parâmetro será atualizado;
`subtoolName` - o nome da subferramenta;
`string value` - o valor a ser aplicado.

- `public void Measure(string technique_string, SequenceSpec sequenceSpec):`

Esta função inicia uma medição dentro do conjunto de parâmetros estabelecido para uma técnica de medição. Dependendo da técnica e dos parâmetros utilizados, esta medição pode durar um tempo finito ou continuar indefinidamente. Um chamado para `IsMeasuring()` pode ser usado para verificar se a medição está completa. Os resultados dos gráficos são atualizados durante a medição e podem ser recuperados periodicamente com chamadas para `GetGraphData()`.

Parâmetros:

`technique_string` - Uma string que indica a técnica a ser usada.
`sequenceSpec` - Uma instância de um conjunto de parâmetros que será usado na técnica de medição.

- `public bool IsMeasuring():`

Uma chamada para `IsMeasuring()` pode ser usada para indicar quando uma medição está completa.

- `public void AbortMeasurement():`

Uma chamada para `AbortMeasurement()` pode ser usada para abortar uma medição em andamento.

- `public GraphData GetGraphData(string plottype):`

Esta função recupera os dados usados para popular a janela de gráficos dentro da interface gráfica do `SensorPal`.

Inclui:

`plottype` - Uma string indicando o tipo de gráfico.

Dados do gráfico incluem:

`InfoPaneNames[00..03]` - Os rótulos de até quatro painéis de informação.

InfoPaneValues[00..03] - Os valores de até quatro painéis de informação.

XAxisLegend - A legenda do eixo X.

XAxisSeries[00..lastResult] - Os valores da série do eixo X.

YAxesLegends[00..lastYAxis] - Legendas para pelo menos um eixo Y.

YAxesSeries[00..lastYAxis][00..lastResult] - Série de dados para pelo menos um eixo Y.

Com essas funções encapsulou-se o controle do equipamento de EIS em módulo Python para usar posteriormente junto com os demais equipamentos. O código implementado é listado abaixo:

```
import numpy as np
import pandas as pd
import serial.tools
import clr
import sys

def get_data_as_df(graph_data):
    dict_ = dict()
    for field_index in range(0, len(graph_data.InfoPaneNames)):
        if graph_data.InfoPaneNames[field_index]:
            try:
                value = float(graph_data.InfoPaneValues[field_index])
            except:
                value = graph_data.InfoPaneValues[field_index]

            dict_[graph_data.InfoPaneNames[field_index]] = value

    return pd.DataFrame(data=dict_, index=[0])

class NoConnectionException(Exception):
    """The device is not connected! Verify the COM port"""

class EIS:
    def __init__(
        self,
        com_port="COM8",
        sensorpal_installation=\\
            "C:\\Program Files (x86)\\Analog Devices\\SensorPal",
        start_frequency="100",
        stop_frequency="1000",
        points="-1",
        dc_bias="1200",
        ac_amplitude="300",
        logarithmic="1",
        calibration_resistor="50"
```

```
) -> None:

self.com_port = com_port
self.sensorpal_installation = sensorpal_installation
self.start_frequency = start_frequency
self.stop_frequency = stop_frequency
self.points=points
self.dc_bias=dc_bias
self.ac_amplitude=ac_amplitude
self.logarithmic=logarithmic
self.calibration_resistor=calibration_resistor

sys.path.append('%s' % self.sensorpal_installation)

clr.AddReference('SensorPal.API')
from SensorPal.API import API as SensorPalAPI

self.sensorpal = SensorPalAPI()

try:
    self.sensorpal.OpenConnection(com_port)
except Exception as e:
    print(e)
    print("I was unable to open '%s'. Is it connected?" %
          com_port)
else:
    self.technique = "Battery Impedance"
    self.technique_parameters =\
        self.sensorpal.GetDefaultTechniqueParameters(self.
            technique)

    self.sensorpal.UpdateTechniqueParameter(
        self.technique,
        self.technique_parameters,
        "Start Frequency",
        self.start_frequency
    )
    self.sensorpal.UpdateTechniqueParameter(
        self.technique,
        self.technique_parameters,
        "Stop Frequency",
        self.stop_frequency
    )
    self.sensorpal.UpdateTechniqueParameter(
        self.technique,
        self.technique_parameters,
        "Points", self.points
```



```

    )
    self.sensorpal.UpdateTechniqueParameter(
        self.technique,
        self.technique_parameters,
        "DC Bias", self.dc_bias
    )
    self.sensorpal.UpdateTechniqueParameter(
        self.technique, self.technique_parameters,
        "AC Amplitude", self.ac_amplitude
    )
    self.sensorpal.UpdateTechniqueParameter(
        self.technique, self.technique_parameters,
        "Logarithmic", self.logarithmic
    )
    self.sensorpal.UpdateTechniqueParameter(
        self.technique,
        self.technique_parameters,
        "Calibration Resistor", self.calibration_resistor
    )
    self.sensorpal.UpdateTechniqueParameter(
        self.technique,
        self.technique_parameters,
        "Enable", "1"
    )

def run(self, filename):
    try:
        self.sensorpal.Measure(self.technique, self.
            technique_parameters)
    except Exception as e:
        print(e)
        raise NoConnectionException
    else:
        dfi = []
        while self.sensorpal.IsMeasuring():
            graph_data = self.sensorpal.GetGraphData("Nyquist")
            if graph_data:
                dfi.append(get_data_as_df(graph_data))

        self.sensorpal.CloseConnection()
        df = pd.concat(dfi, ignore_index=True)
        df.to_csv(filename)

```

Listagem 3.1 – Módulo Python implementado para o equipamento de EIS.

4 Carga eletrônica DC

A carga eletrônica usada nesse projeto foi a Keysight EL34243A ([Keysight Technologies, 2024c](#)).

4.1 Características da EL34243A

As principais características da EL34243A são ([Keysight Technologies, 2024a](#))

- Carga limpa e confiável
 - 250 a 600 W, 0 a 150 V, 0 a 60 A
 - Quatro modos: tensão constante (CV), corrente constante (CC), potência constante (CP) e resistência constante (CR)
 - Excelente precisão de programação/leitura
 - Alta resolução
 - Sense remoto de 2 ou 4 fios
 - Proteção contra sobretensão, sobrecorrente e sobreaquecimento
- Capacidades convenientes para bancada
 - Resposta transiente dinâmica até 100 kHz
 - Velocidade do ventilador controlada termicamente para menor ruído acústico
 - Conexões paralelas de entrada (somente EL34243A)
 - Terminal de entrada frontal e traseiro
- Interfaces intuitivas e fáceis de usar
 - Display colorido de LCD de 4,3 polegadas
 - Canais codificados por cores
 - Botões individuais para entrada de tensão e corrente
 - LAN (LXI), USB e GPIB (opcional)
- Caracterização avançada
 - Visualização do escopo
 - Registro de dados
 - Sequenciamento de entrada

Figura 15 – Detalhes dos equipamentos da série EL30000, da Keysight.

Feature	EL33133A	EL34143A	EL34243A
Power input	250 W	350 W	300 W per input, total 600 W
IO interface	USB	USB/LAN standard, GPIB optional	USB/LAN standard, GPIB optional
Maximum operating voltage/current	150 V/40 A	150 V/60 A	150 V/60 A
Digital IO		✓	✓
Scope view	✓	✓	✓
Input-parallel mode			✓
Input sequencing		✓	✓
Low range current measurement		✓	✓
Data logging	✓	✓	✓
Battery testing	✓	✓	✓
BenchVue software	✓	✓	✓

Fonte: ([Keysight Technologies, 2024a](#)).

- Medição de corrente de baixa amplitude
- Taxa de variação de tensão ajustável
- Teste de bateria

Informações específicas sobre o modelo utilizado seguem na figura 15.

4.2 Programação e captura dos dados

O EL34243A possui comunicação serial e aceita comandos VISA. O VISA (Virtual Instrument Software Architecture) da National Instruments é uma API padrão industrial usada para comunicação com instrumentos de teste e medição em uma variedade de plataformas de hardware e sistemas operacionais. Ela possibilita a comunicação através de diversas interfaces, como GPIB, USB, Ethernet, entre outras, permitindo que desenvolvedores e engenheiros controlem dispositivos de hardware e organizem dados de medição de forma eficiente. Essa arquitetura é amplamente adotada devido à sua flexibilidade e interoperabilidade com uma grande variedade de instrumentos e softwares.

A implementação do VISA para comunicação via USB incorpora diversos elementos para assegurar uma interação eficiente com dispositivos de instrumentação. Tecnicamente, o VISA sobre USB pode operar com diferentes tipos de transferência USB: *control*, *bulk* e *interrupt*:

- **Transferências de Controle:** Utilizadas para enviar comandos simples ou solicitar configurações do dispositivo.

- **Transferências em Massa:** Escolhidas para a transmissão de grandes volumes de dados, onde a confiabilidade é mais importante que a velocidade.
- **Transferências por Interrupção:** Usadas para pequenas quantidades de dados que necessitam resposta rápida, como interrupções ou dados de estado.

Essas capacidades são empregadas ao desenvolver drivers de dispositivo USB com VISA, permitindo a implementação da comunicação de dados sem exigir conhecimento detalhado sobre as camadas de comunicação de baixo nível. O envio e a recepção de dados podem ser feitos utilizando comandos padrões do VISA, que são consistentes independentemente da interface física usada.

Os dados são frequentemente transmitidos como strings ASCII através das interfaces VISA, embora o formato possa variar de acordo com as especificações do instrumento e as necessidades do usuário. Isso facilita a comunicação com uma vasta gama de equipamentos de forma padronizada.

Dadas essas informações, estudou-se a programação do EL34243A através do VISA, baseando-se no manual de programação do equipamento ([Keysight Technologies, 2024b](#)). Usando a biblioteca `pyserial`, buscou-se implementar em alto nível o controle desse equipamento para que os seguintes objetivos fossem atingidos

- Configuração de uma lista temporizada de descarga para os dois canais;
- Inicialização e interrupção de uma descarga listada;
- Configuração do *logging* de dados com as leituras dos sensores de corrente e tensão da carga DC por canal;
- Recuperação dos dados salvos via interface serial.

4.2.1 Linguagem SCPI

O padrão de comandos da EL34243A segue a estrutura do SCPI. O Comandos Padrão para Instrumentos Programáveis (*Standard Commands for Programmable Instruments* - SCPI) é uma linguagem de comandos para instrumentos baseada em ASCII, projetada para equipamentos de teste e medição.

Os comandos SCPI são baseados em uma estrutura hierárquica, também conhecida como um sistema em árvore. Neste sistema, comandos associados são agrupados sob um nó ou raiz comum, formando assim subsistemas. Uma parte do subsistema SOURCE é mostrada abaixo para ilustrar o sistema de árvore.

```
[SOURCE:]  
CURRENT <corrente> | MIN | MAX  
CURRENT? MIN | MAX  
CURRENT:  
TRIGgered <corrente> | MIN | MAX  
TRIGgered? MIN | MAX  
VOLTage <voltagem> | MIN | MAX  
VOLTage? MIN/MAX  
VOLTage:  
TRIGgered <voltagem> | MIN | MAX  
TRIGgered? MIN | MAX
```

SOURce é a palavra-chave raiz do comando, CUREnt e VOLTage são palavras-chave de segundo nível, e TRIGgered é a palavra-chave de terceiro nível. Um dois-pontos (:) separa uma palavra-chave de comando de uma palavra-chave de nível inferior.

4.2.1.1 Formato dos comandos

O formato usado para exibir comandos no manual ([Keysight Technologies, 2024b](#)) é mostrado abaixo:

```
CURRent <corrente> | MINimum | MAXimum
```

A sintaxe de comando mostra a maioria dos comandos (e alguns parâmetros) como uma mistura de letras maiúsculas e minúsculas. As letras maiúsculas indicam a grafia abreviada para o comando. Para linhas de comando mais curtas, prefira enviar a forma abreviada.

Por exemplo, na declaração de sintaxe acima, CURR e CURRENT são ambas formas aceitáveis. Pode-se usar maiúsculas ou minúsculas. Portanto, CURRENT, curr, e Curr são todas aceitáveis. Outras formas, como CUR e CURREN, gerarão um erro.

Um caractere de barra vertical (|) separa múltiplas escolhas de parâmetros para uma string de comando.

Chaves angulares (<>) indicam que se deve especificar um valor para o parâmetro incluído. Por exemplo, a declaração de sintaxe mostra o parâmetro de corrente envolvido em chaves angulares. As chaves não são enviadas com a string de comando. Deve-se especificar um valor para o parâmetro (como CURR 0.1).

Chaves ({}) indicam parâmetros que podem ser repetidos zero ou mais vezes. É usado especialmente para mostrar arrays. A notação {A},{B}* mostra que o parâmetro

"A" deve ser inserido, enquanto o parâmetro "B" pode ser omitido ou pode ser inserido uma ou mais vezes.

Alguns parâmetros são colocados entre colchetes ([]). Os colchetes indicam que o parâmetro é opcional e pode ser omitido. Os colchetes não são enviados com a string de comando. Se for especificado um valor para um parâmetro opcional, o instrumento escolhe um valor padrão.

Um cólon (:) separa uma palavra-chave de comando de uma palavra-chave de nível inferior. Deve-se inserir um espaço em branco para separar um parâmetro de uma palavra-chave de comando. Se um comando requer mais de um parâmetro, deve-se separar parâmetros adjacentes usando uma vírgula como mostrado abaixo:

```
APPLY CH1,3.5,1.5
```

4.2.1.2 Separadores de comando

Dois pontos (:) separam uma palavra-chave de comando de uma palavra-chave de nível inferior, como mostrado abaixo:

```
SOURCE:CURRENT:TRIGgered
```

Um ponto e vírgula (;) é usado para separar dois comandos dentro do mesmo subsistema, e também pode minimizar a digitação. Por exemplo, enviar a seguinte string de comando:

```
SOUR:VOLT MIN;CURR MAX
```

é o mesmo que enviar os seguintes dois comandos:

```
SOUR:VOLT MIN
```

```
SOUR:CURR MAX
```

Usa-se um dois pontos e um ponto e vírgula para vincular comandos de diferentes subsistemas. Por exemplo, na seguinte string, um erro é gerado se não se usar o dois pontos e ponto e vírgula:

```
DISP:TEXT:CLE;SOUR:CURR MIN
```

4.2.1.3 Terminadores de comando

Uma cadeia de comandos enviada para o instrumento deve ser finalizada com um caractere de *nova linha*. A norma IEEE-488 EOI (end-of-identity) interpreta este como um caractere de *nova linha* e pode ser utilizado para finalizar uma cadeia de comandos em substituição ao caractere de *nova linha* padrão. A combinação de um *retorno de carro* seguido por uma *nova linha* também é aceita. O término da cadeia de comandos sempre redefinirá o caminho do comando SCPI atual para o nível raiz. O caractere *nova linha* possui o código decimal ASCII de 10.

4.2.1.4 Comandos comuns ao IEEE-488.2

A norma IEEE-488.2 define um conjunto de comandos comuns que executam funções como reinicialização, auto-teste e operações de status. Os comandos comuns sempre começam com um asterisco (*), possuem de quatro a cinco caracteres de comprimento e podem incluir um ou mais parâmetros. A palavra-chave do comando é separada do primeiro parâmetro por um espaço em branco. Utiliza-se um ponto e vírgula (;) para separar múltiplos comandos como mostrado a seguir:

```
*RST;*CLS;*ESE 32;*OPC?
```

4.2.1.5 Tipos de Parâmetros SCPI

A linguagem SCPI define vários formatos de dados diferentes para serem usados em mensagens de programa e mensagens de resposta.

4.2.1.6 Parâmetros de Canal

O parâmetro de canal <chanlist> é necessário para endereçar um ou mais canais. Possui a seguinte sintaxe:

```
(@<canal> | ,<canal> | ,<canal> | ,<canal>)
```

Também é possível especificar uma sequência de canais da seguinte forma:

```
(<canal inicial>:<canal final>)
```

Por exemplo, (@2) especifica o canal 2 e (@1:3) especifica os canais 1 a 3. A lista de canais, mostrada como <chanlist> neste documento, deve ser precedida pelo símbolo @ e deve estar entre parênteses. Um máximo de 4 canais pode ser especificado por meio de uma combinação de canais individuais e intervalos. Os resultados das consultas são retornados na ordem em que são especificados na lista.

Nota: Ao adicionar uma lista de parâmetros de canal a uma consulta, deve-se incluir um espaço entre o indicador de consulta (?) e o último parâmetro da lista de canais. Caso contrário, ocorrerá o erro -103, Separador Inválido.

4.2.1.7 Parâmetros Numéricos

Comandos que requerem parâmetros numéricos aceitarão todas as representações decimais comuns de números, incluindo sinais opcionais, pontos decimais e notação científica. Valores especiais para parâmetros numéricos como MINimum, MAXimum e DEFault também são aceitos.

Também é possível enviar sufixos (V, A ou SEC) com parâmetros numéricos. Se apenas valores numéricos específicos forem aceitos, a carga eletrônica arredondará automaticamente os parâmetros numéricos de entrada. O seguinte comando usa um parâmetro numérico:

```
CURR <corrente> | MIN | MAX
```

4.2.1.8 Parâmetros Discretos

Parâmetros discretos são usados para programar configurações que possuem um número limitado de valores, como BUS e IMM. As respostas das consultas sempre retornam a forma curta em letras maiúsculas. O seguinte comando usa parâmetros discretos:

```
TRIG:SOUR BUS | IMM
```

4.2.1.9 Parâmetros Booleanos

Parâmetros booleanos representam uma única condição binária que é verdadeira ou falsa. Para uma condição falsa, o instrumento aceitará OFF ou 0. Para uma condição verdadeira, o instrumento aceitará ON ou 1. Quando se consulta um parâmetro booleano, o instrumento sempre retornará 0 ou 1. O seguinte comando usa um parâmetro booleano:

```
DISP OFF | ON
```

4.2.2 Comandos de interesse

Alguns dos comandos de interesse para o contexto desse projeto são listados a seguir

- *IDN?

A consulta retorna a string de identificação do instrumento. Um exemplo de retorno é

Keysight Technologies,EL34243A,MY00000001,X.X.X-X.X.X-X-X

Os quatro campos separados por vírgulas são o nome do fabricante, o número do modelo, o número serial e o código de revisão. O primeiro "X.X.X" no código de revisão é o número de revisão do firmware do controlador; o segundo é do firmware do painel frontal do equipamento; o terceiro é da FPGA primária e o quarto é da FPGA secundária do equipamento;

- ***RST**

Este comando reseta o instrumento para as configurações default de quando ele é ligado e aborta a execução de qualquer outro comando ou processo ativo no equipamento;

- ***OPC?**

Este comando é usado para verificar se a última operação executada pelo instrumento foi completada;

- **SENSe:DLOG:FUNCtion:VOLTage ON | OFF | 1 | 0[, (@<chanlist>)]**
& SENSe:DLOG:FUNCtion:CURREnt ON | OFF | 1 | 0[, (@<chanlist>)]

Estes comandos ligam ou desligam o *logging* de dados dos sensores de tensão ou corrente nos canais especificados;

- **FUNCtion CURREnt | VOLTage | POWer | RESistance[, (@<chanlist>)]**

Esse comando configura o modo de operação dos canais listados;

- **LIST:COUNt <count> | MINimum | MAXimum | INFINITY[, (@<chanlist>)]**

Configura valores de amplitude numa lista de descarga para os canais especificados;

- **LIST:DWELL <value>,<value >[, (@<chanlist>)]**

Esse comando especifica o tempo de duração de cada patamar do comando anterior na execução da lista de descarga;

- **FETC:DLOG? 1, (@channels)**

Esse comando solicita uma amostra dos dados armazenados no logging.

Para que o objetivo de programar, executar e ler dados de uma descarga listada seja alcançado, mais comandos de configuração são necessário. No entanto esses comandos

adicionais são deixados de fora desse documento por estarem bem documentados no manual do fabricante ([Keysight Technologies, 2024b](#)). De posse dos comandos SCPI necessários para executar a lista na carga DC, implementou-se em Python um módulo para controlar o equipamento. A listagem desse módulo segue abaixo.

```
import pyvisa
import time
import pandas as pd

class DCL:
    def __init__(self, serial_number="USB0::0x2A8D::0x3902::MY61001566::INSTR") -> None:
        self.rm = pyvisa.ResourceManager()
        self.inst = self.rm.open_resource(serial_number)
        print(self.inst.query("*IDN?"))

    def reset(self):
        while int(self.inst.query("*RST;*OPC?")) != 1:
            pass

        self.inst.write("SENS:DLOG:FUNC:VOLT 0, (@1,2)")
        self.inst.write("SENS:DLOG:FUNC:CURR 0, (@1,2)")

    def run(self, mode, list_values, channels,
            list_time=[0.01], repetitions=1, sample_time=0.01, filename="DCL.csv"):

        self.inst.write(f"FUNC {mode}, (@{channels})")
        self.inst.write(f"{mode} 0.012, (@{channels})")
        self.inst.write(f"LIST:{mode} {''.join([str(e) + ',' for e in list_values])} (@{channels})")

        npad = len(list_values) - len(list_time)
        time_list = list_time.copy()
        time_list.extend(npad * [list_time[-1]])

        nlist = len(list_values)
        eost_values = [0] * (nlist - 1) + [1]

        self.inst.write(f"LIST:DWEL {''.join([str(e) + ',' for e in time_list])} (@{channels})")
        self.inst.write(f"LIST:COUNT {repetitions}, (@{channels})")
        self.inst.write(f"LIST:TERM:LAST 0, (@{channels})")
        self.inst.write(f"LIST:STEP AUTO, (@{channels})")
        self.inst.write(f"{mode}:MODE LIST, (@{channels})")
```

```

self.inst.write(f"LIST:TOUT:EOST {' '.join([str(e) for e in
    eost_values])}, (@{channels})")

self.inst.write(f"SENS:DLOG:FUNC:VOLT 1, (@1,2)")
self.inst.write(f"SENS:DLOG:FUNC:CURR 1, (@1,2)")

self.inst.write(f"SENS:DLOG:TIME {sum(time_list)}")
self.inst.write(f"SENS:DLOG:PER {sample_time}")

self.inst.write(f"DIG:PIN1:FUNC TOUT")
self.inst.write(f"DIG:PIN1:POL NEG")

self.inst.write("TRIG:TRAN:SOUR BUS")
self.inst.write("TRIG:DLOG:SOUR BUS")

self.inst.write(f"INIT:TRAN (@{channels})")
self.inst.write(f'INIT:DLOG "Internal:/log1.dlog"')

self.inst.write(f"INP ON, (@{channels})")
self.inst.write("*TRG")

time.sleep(sum(time_list) + 5)

fetch_points = int(sum(time_list) / sample_time)
dfi = []
for i in range(fetch_points):
    dout = self.inst.query(f"FETC:DLOG? 1, (@{channels})")
    dout_float = [float(a) for a in dout.split(",")]
    dfi.append(
        pd.DataFrame(data={"Voltage": dout_float[0], "Current":
            dout_float[1]}, index=[0]))

df = pd.concat(dfi, ignore_index=True)
df.to_csv(filename)

return df

```

Listagem 4.1 – Módulo Python implementado para a carga DC.

5 Composição do Sistema

Uma vez que o controle dos equipamentos de EIS e a carga DC estavam programados e interfaceados com o usuário final na forma de módulos Python; projetou-se um esquema de chaveamento entre os equipamentos que permitisse realizar descargas e realizar ensaios de espectroscopia de maneira alternada e automática. Para isso fez-se necessário o uso de uma placa de relés de 5V e de um microcontrolador para comandar essa placa.

5.1 Arquitetura do Sistema

O sistema foi montado segundo o diagrama arquitetural mostrado na figura 16.

5.2 Conexão entre os equipamentos

5.2.1 Conexão elétrica

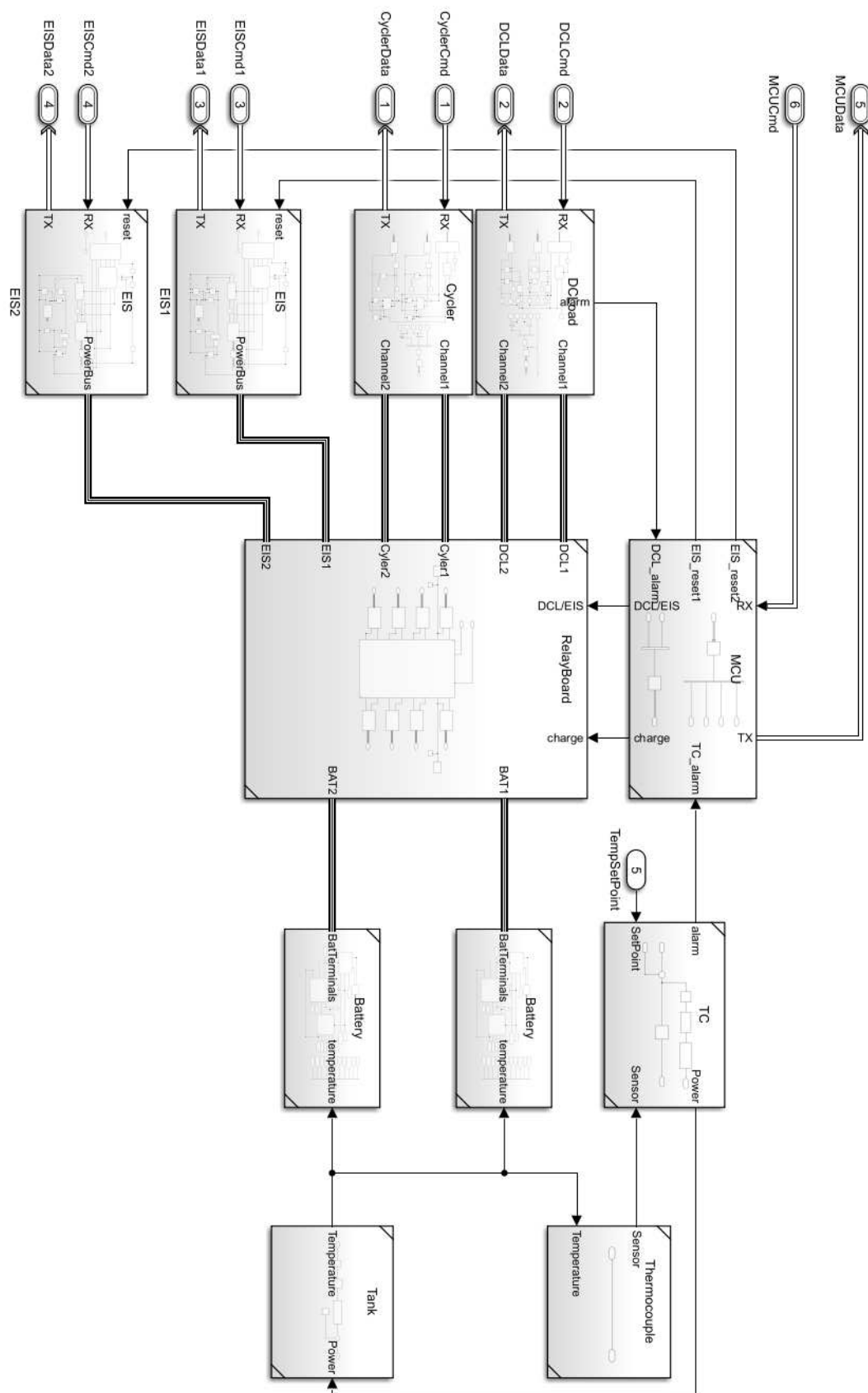
Dada a disponibilidade de dois canais da carga DC, projetou-se o sistema para trabalhar com duas baterias por vez, o que demandou o uso de dois equipamentos de espectroscopia. Usou-se uma placa de 8 relés e 5V, com corrente máxima de 10A cada. A placa em questão aparece na figura . A conexão das baterias, da carga DC e dos equipamentos de EIS seguem na figura 18.

Como pode ser visto na figura 18, uma vez que se deseja realizar os mesmos ensaios simultaneamente para as duas baterias conectadas, apenas dois sinais de comando para o relé são necessários. O primeiro sinal (Charge) faz o chaveamento das baterias entre os equipamentos de caracterização e descarga e o carregador das baterias. Já o segundo sinal (DCLoad/EIS) faz o chaveamento entre a carga DC e os equipamentos de espectroscopia.

5.2.2 Conexão de dados

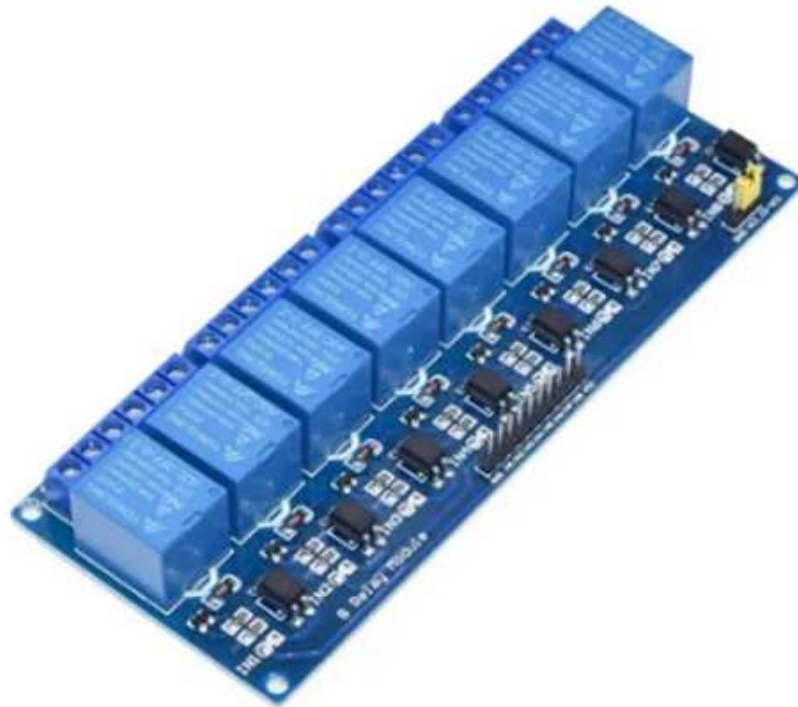
Todos os equipamentos do sistema se comunicam com o computador via USB. Para possibilitar que todos se conectem ao mesmo computador, utilizou-se um Hub USB para centralizar as conexões dos cabos num único ponto. A imagem do hub utilizado segue na figura 19.

Figura 16 – Arquitetura da plataforma de caracterização de baterias.



Fonte: Autoria própria.

Figura 17 – Placa de relés com 8 canais.



Fonte: (Bit Maker, 2024).

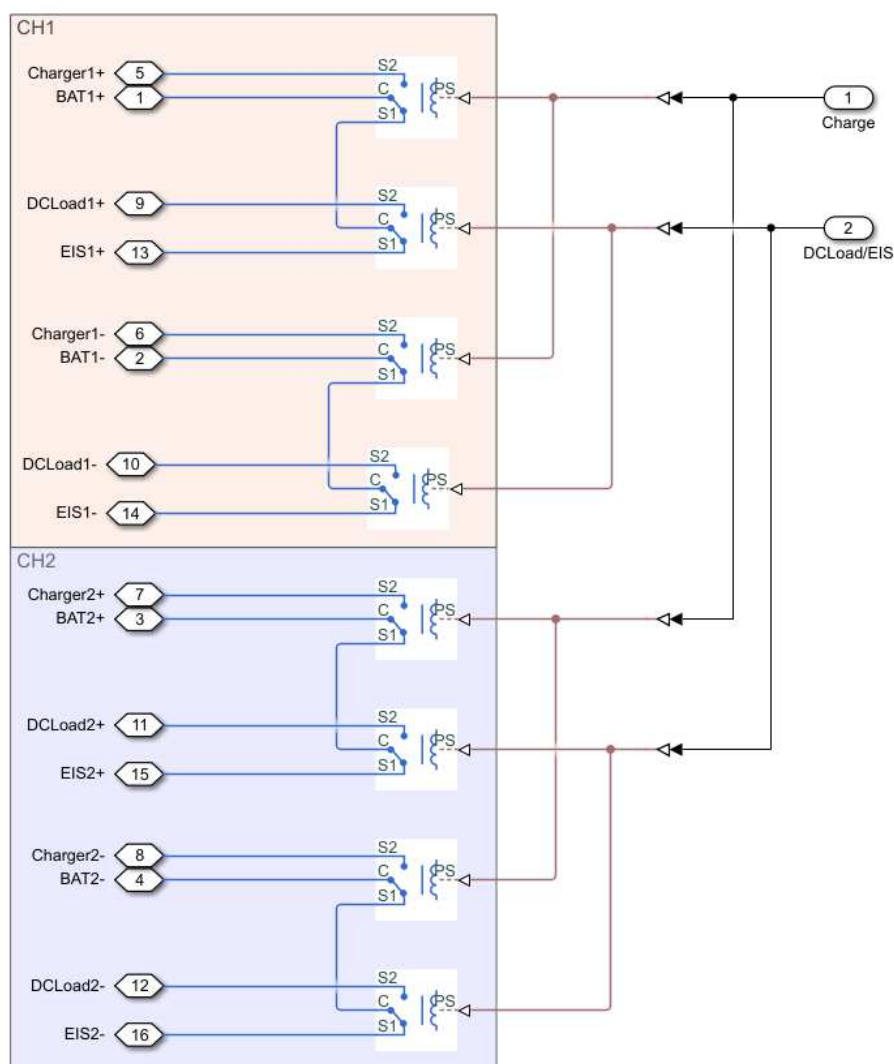
5.3 Controle da placa de relés

O controle da placa de relés foi feito com a plataforma de desenvolvimento micro-controlada ESP32-Devkit V1, que aparece na figura 20.

Uma vez que os equipamentos precisam estar bem fixados numa plataforma para evitar desconexões e acidentes, projetou-se um *shield* para a ESP, para que a conexão com a placa de relés pudesse ser feita através de conectores de borda. Nesse *shield* fez-se necessária a inclusão de deslocadores de nível lógico para que a ESP pudesse comandar os relés de 5V com os sinais de 3.3V produzidos por seus pinos digitais. O esquemático desse *shield* segue na figura 21.

O *shield* projetado contém algumas saídas e entradas adicionais fora as de comando dos relés. Os dois sinais de saída correspondem ao comando de reset de cada placa de espectroscopia. Já os sinais de entrada são para recebimento dos sinais de *trigger* da carga DC e do controlador de temperatura que possivelmente venha a ser utilizado numa versão futura da plataforma de caracterização. A inclusão desses sinais de entrada dá flexibilidade para o sistema operar baseado em eventos, onde se programa a carga e o controlador de temperatura para informar diretamente a ESP sobre o estado da câmara térmica (temperatura alvo atingida ou não) e a conclusão de descargas por parte da carga DC. A PCB projetada segue na figura 22. Já o resultado final consta nas figuras 23 e 24.

Figura 18 – Conexão dos equipamentos à placa de relés.



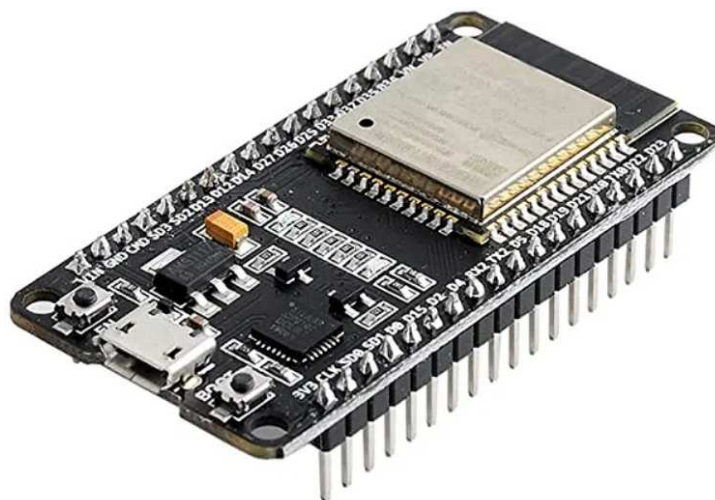
Fonte: Autoria própria.

Figura 19 – Hub USB utilizado.

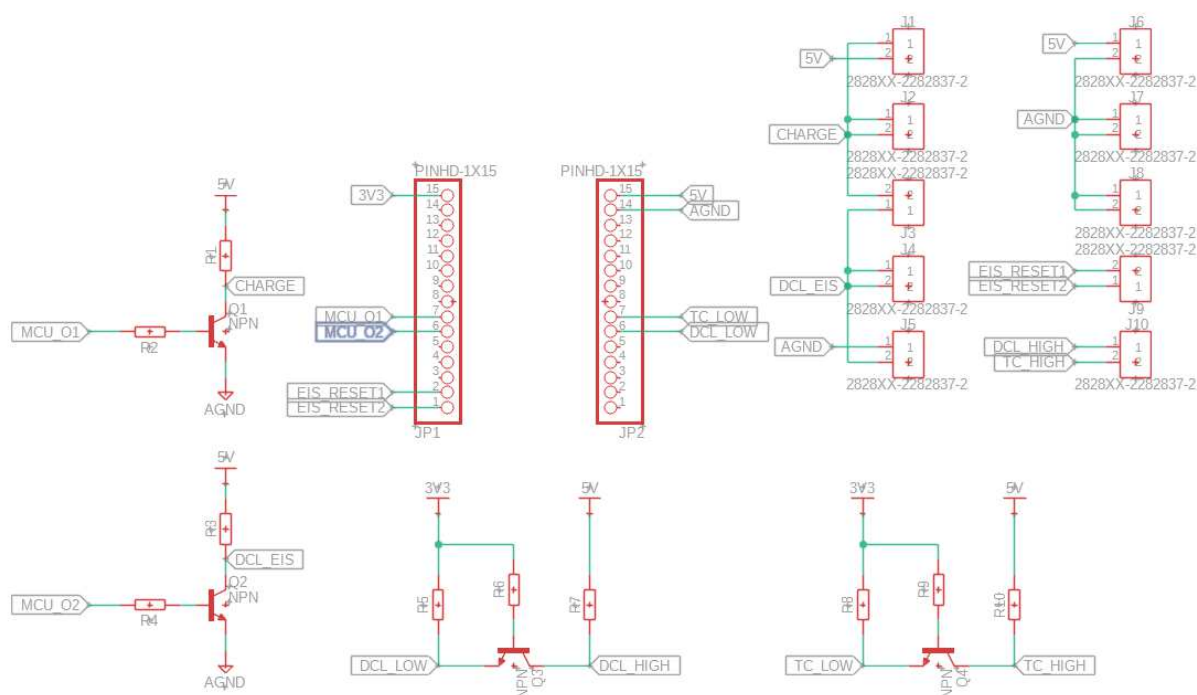


Fonte: Autoria própria.

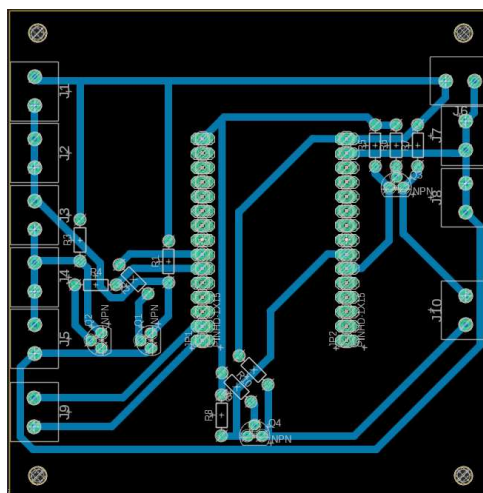
Figura 20 – ESP32-Devkit V1.



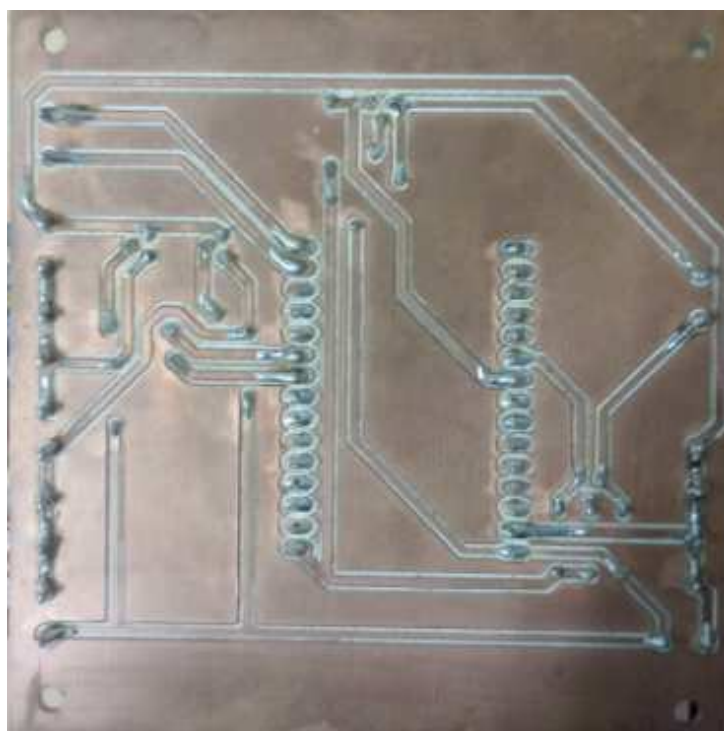
Fonte: (IOT Robótica, 2024).

Figura 21 – Esquemático do *shield* para a ESP.

Fonte: Autoria própria.

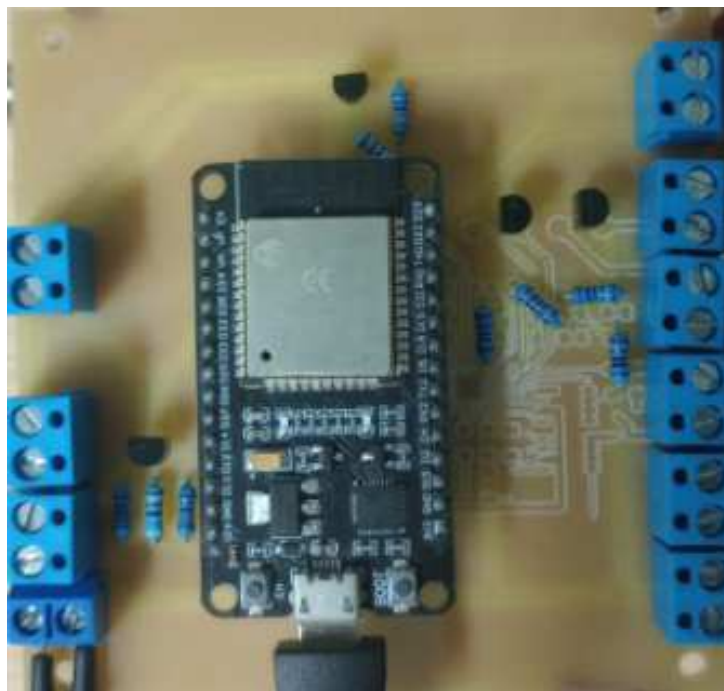
Figura 22 – Desenho da PCB do *shield* da ESP.

Fonte: Autoria própria.

Figura 23 – Vista inferior da PCB do *shield* da ESP.

Fonte: Autoria própria.

Figura 24 – Vista superior da PCB do *shield* da ESP.



Fonte: Autoria própria.

5.3.1 Firmware da ESP32-Devkit V1 e software de comunicação

Para controlar a ESP através da porta serial usando Python precisou-se programar o firmware do microcontrolador. Os comandos ASCII criados são no formato `\;SINAL ESTADO;` para o controle da placa de relés, onde **SINAL** pode ser **CHARGE** ou **DCL** e **ESTADO** pode ser **ON** ou **OFF**. Para o reset das placas de espectroscopia os comandos podem ser `;RESET_EIS1;` ou `;RESET_EIS2;`, onde a numeração indica qual placa resetar. Para a interrogação do estado do controlador de temperatura e da carga DC os comandos são `;TC_ALARM?;` e `;DCL_ALARM?;`, respectivamente. Esses comandos fazem o micro-controlador retornar **ON** ou **OFF**, à depender se o *trigger* em questão está alto ativo ou não. Abaixo segue a listagem do firmware implementado em C para a ESP.

```
#include <stdio.h>
#include <string.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/queue.h"
#include "driver/uart.h"
#include "driver/gpio.h"

#define BUF_SIZE      1024
#define TX_PIN        1
#define RX_PIN        3
#define CHARGE         18
#define DCL_EIS        19
#define EIS_RESET_1    22
#define EIS_RESET_2    23
#define TC_ALARM       33
#define DCL_ALARM      32

static QueueHandle_t uart0_queue;

static void uart_event_task(void *pvParameters) {
    uart_event_t event;
    uint8_t* dtmp = (uint8_t*) malloc(BUF_SIZE);
    uint8_t* buffer = (uint8_t*) calloc(1, BUF_SIZE);
    int buffer_len = 0;

    for(;;) {
        // Espera por dados da UART
        if(xQueueReceive(uart0_queue, (void *)&event, portMAX_DELAY)) {
            bzero(dtmp, BUF_SIZE);
            switch(event.type) {
                case UART_DATA:
                    uart_read_bytes(UART_NUM_0, dtmp, event.size,
                                    portMAX_DELAY);
                    for (int i = 0; i < event.size; ++i) {
```

```

buffer[buffer_len++] = dtmp[i];
if (dtmp[i] == ',') {
    char *token = strtok((char *)buffer, ",");
    if (strcmp(token, "CHARGE ON") == 0) {
        gpio_set_level(CHARGE, 1);
    } else if (strcmp(token, "CHARGE OFF") == 0) {
        {
            gpio_set_level(CHARGE, 0);
        }
    } else if (strcmp(token, "DCL_EIS ON") == 0) {
        {
            gpio_set_level(DCL_EIS, 1);
        }
    } else if (strcmp(token, "DCL_EIS OFF") ==
0) {
        {
            gpio_set_level(DCL_EIS, 0);
        }
    } else if (strcmp(token, "RESET_EIS1") == 0) {
        {
            gpio_set_level(EIS_RESET_1, 1);
            vTaskDelay(500 / portTICK_PERIOD_MS);
            gpio_set_level(EIS_RESET_1, 0);
        }
    } else if (strcmp(token, "RESET_EIS2") == 0) {
        {
            gpio_set_level(EIS_RESET_2, 1);
            vTaskDelay(500 / portTICK_PERIOD_MS);
            gpio_set_level(EIS_RESET_2, 0);
        }
    } else if (strcmp(token, "TC_ALARM?") == 0) {
        {
            char msg[50];
            sprintf(msg, "TC_ALARM %s\n",
                gpio_get_level(TC_ALARM) ? "ON" : "
                OFF");
            uart_write_bytes(UART_NUM_0, msg, strlen
                (msg));
        }
    } else if (strcmp(token, "DCL_ALARM?") == 0) {
        {
            char msg[50];
            sprintf(msg, "DCL_ALARM %s\n",
                gpio_get_level(DCL_ALARM) ? "ON" : "
                OFF");
            uart_write_bytes(UART_NUM_0, msg, strlen
                (msg));
        }
    }
    buffer_len = 0;
}
}
break;
default:
    break;

```

```

        }
    }
}

free(dtmp);
free(buffer);
vTaskDelete(NULL);
}

void app_main() {
    uart_config_t uart_config = {
        .baud_rate = 9600,
        .data_bits = UART_DATA_8_BITS,
        .parity     = UART_PARITY_DISABLE,
        .stop_bits  = UART_STOP_BITS_1,
        .flow_ctrl  = UART_HW_FLOWCTRL_DISABLE
    };
    uart_param_config(UART_NUM_0, &uart_config);
    uart_set_pin(UART_NUM_0, TX_PIN, RX_PIN, UART_PIN_NO_CHANGE,
        UART_PIN_NO_CHANGE);
    uart_driver_install(UART_NUM_0, BUF_SIZE * 2, 0, 20, &uart0_queue,
        0);

    gpio_set_direction(CHARGE, GPIO_MODE_OUTPUT);
    gpio_set_direction(DCL_EIS, GPIO_MODE_OUTPUT);
    gpio_set_direction(EIS_RESET_1, GPIO_MODE_OUTPUT);
    gpio_set_direction(EIS_RESET_2, GPIO_MODE_OUTPUT);
    gpio_set_direction(TC_ALARM, GPIO_MODE_INPUT);
    gpio_set_direction(DCL_ALARM, GPIO_MODE_INPUT);

    xTaskCreate(uart_event_task, "uart_event_task", 2048, NULL, 12, NULL
        );
}

```

Listagem 5.1 – Firmware do micro-controlador ESP.

Para manter o padrão dos módulos desenvolvidos até então, implementou-se também um módulo em Python para comunicação com a ESP. A listagem dessa implementação segue abaixo.

```

import serial
import time

class MCU:
    def __init__(self, com_port="COM3", baud=9600):
        self.serial = serial.Serial(com_port, baud)

    def close(self):
        self.serial.close()

```

```
def is_open(self) -> bool:
    return self.serial.is_open

def query_alarm(self, equip:str) -> bool:
    if equip in ['TC', 'DCL']:
        self.serial.write(f'{equip}_ALARM?', '.encode()')
        resposta = self.serial.readline().decode().strip()

        if resposta == f"{equip}_ALARM ON":
            return True
        elif resposta == f"{equip}_ALARM OFF":
            return False
    else:
        return False

def reset_eis(self, eis_ch:int):
    self.serial.write(f'RESET_EIS{eis_ch}', '.encode()')

def cycler(self, onoff:bool):
    self.serial.write(f'CHARGE {"ON" if onoff else "OFF"}', '.encode()')

def dcl_eis(self, onoff:bool):
    self.serial.write(f'DCL_EIS {"ON" if onoff else "OFF"}', '.encode()')
```

Listagem 5.2 – Módulo em Python para comunicação com a ESP.

5.4 Montagem

O sistema foi montado sobre uma placa de madeira, onde os equipamentos foram fixados, com exceção da carga DC, que ficou sobre a bancada. A montagem segue na figura [25](#).

Figura 25 – Plataforma de caracterização de baterias.



Fonte: Autoria própria.

6 Câmara Térmica

Projetou-se uma câmara térmica de pequeno porte para realizar a caracterização das baterias em um ambiente com temperatura controlada. A câmara ainda está em processo de confecção.

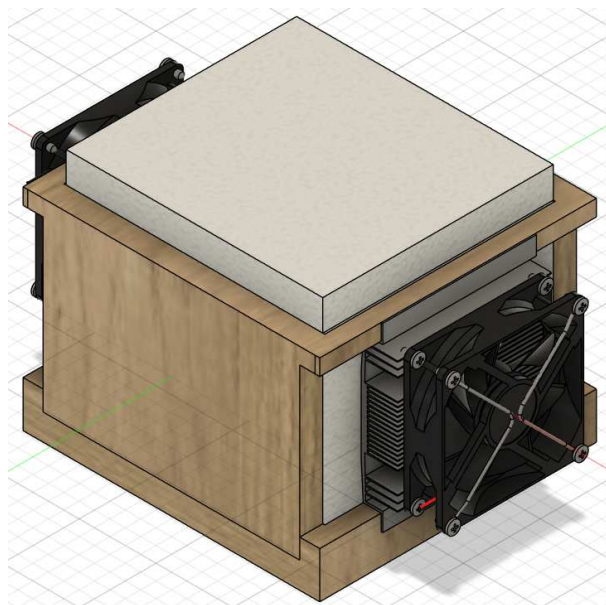
6.1 Modelagem 3D

Criou-se um modelo 3D da câmara térmica no software de modelagem 3D Fusion 360. O projeto focou em uma câmara individual para cada bateria, como forma de diminuir a massa térmica do volume de ar interno e evitar que as baterias troquem calor entre si à medida que a descarga faça suas temperaturas internas aumentarem. O modelo criado segue nas figuras 26, 27 e 28.

A câmara consiste de uma caixa de poliestireno extrudado encaixada sobre um suporte de MDF e com dois atuadores. O atuador consiste em duas placas de Peltier empilhadas com dois dissipadores, uma em cada Peltier. Cada dissipador possui um cooler. Portanto, têm-se para cada atuador um cooler que fica na parte interna da caixa e outro que fica na parte externa. As propriedades de cada elemento dessa caixa constam na tabela 1 e foram obtidos da geometria dos dispositivos escolhidos e dados do datasheet.

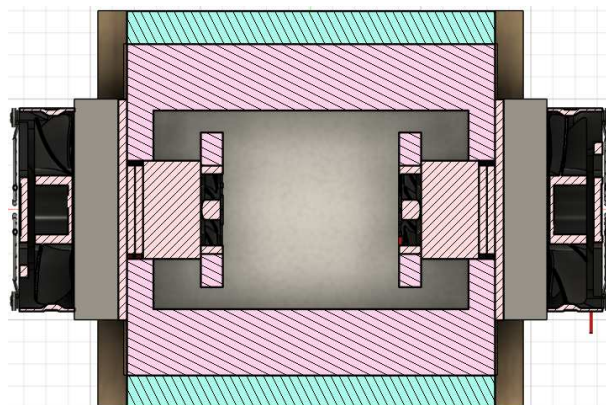
De posse das informações da tabela 1 foi possível construir o modelo da câmara no ambiente de simulação do Simulink. Para simular a dinâmica térmica no interior da

Figura 26 – Modelo da câmara térmica no Fusion 360.



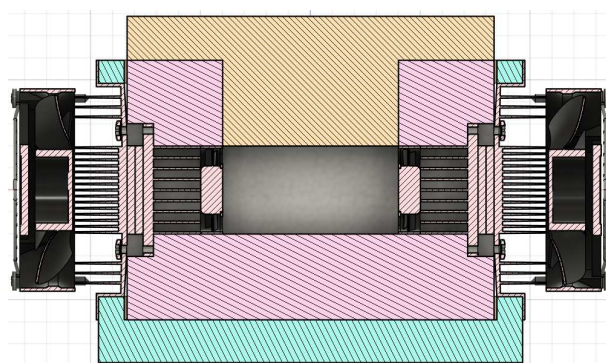
Fonte: Autoria própria.

Figura 27 – Vista superior em corte da câmara térmica no Fusion 360.



Fonte: Autoria própria.

Figura 28 – Vista frontal em corte da câmara térmica no Fusion 360.



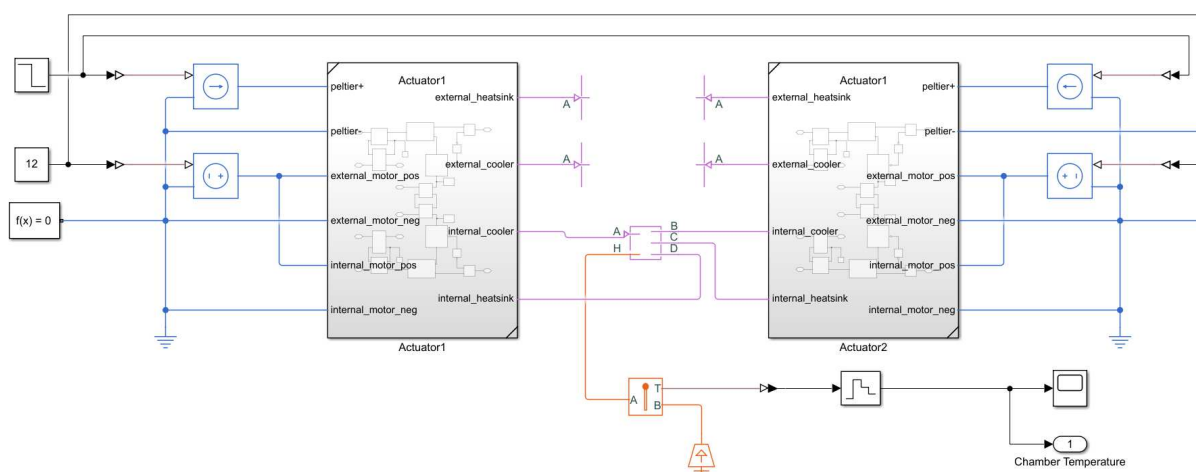
Fonte: Autoria própria.

câmara modelou-se o volume interno como uma massa de ar termicamente isolada do exterior. As dimensões internas da câmara são 80x90x50mm. O modelo construído segue nas figuras 29 e 30.

Simulou-se o modelo para correntes de 1A hora fluindo em um sentido hora em outro. Os resultados constam nas figuras 31 e 32.

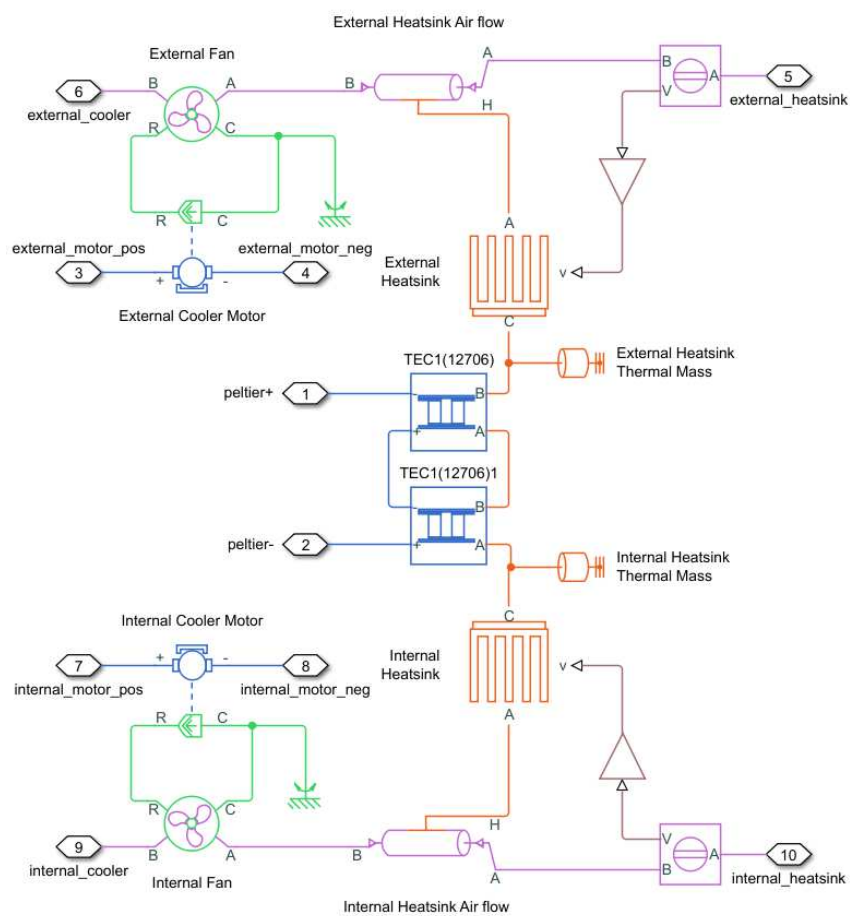
Com base nesses resultados é possível projetar um controle de temperatura para a câmara térmica. Essa atividade não chegou a ser desenvolvida no estágio.

Figura 29 – Modelo da câmara térmica no Simulink.



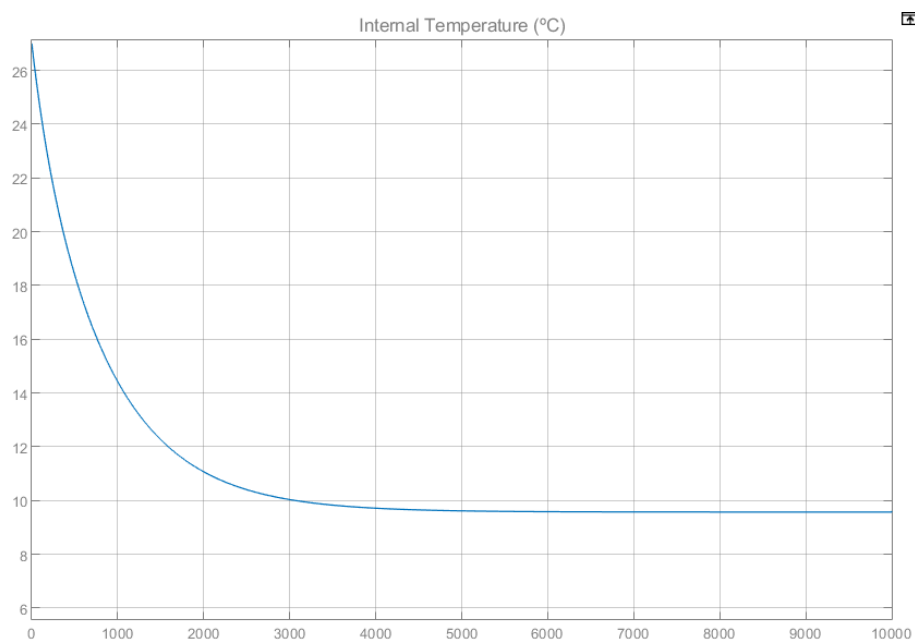
Fonte: Autoria própria.

Figura 30 – Modelo do atuador da câmara térmica no Simulink.



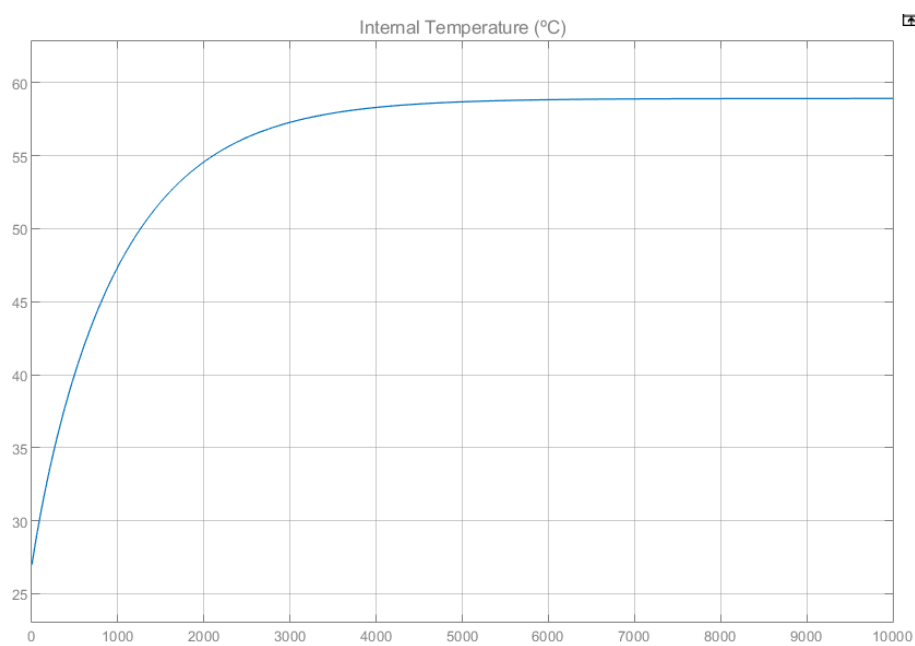
Fonte: Autoria própria.

Figura 31 – Resfriamento simulado da câmara térmica.



Fonte: Autoria própria.

Figura 32 – Aquecimento simulado da câmara térmica no Simulink.



Fonte: Autoria própria.

Tabela 1 – Parâmetros dos Dispositivos da Câmara.

Propriedade	Interno	Externo
Dissipador		
Massa da Base (kg)	0.3248	0.372
Área da Base (m ²)	0.0025	0.010816
Calor Específico (J/kgK)	900	900
Volume (m ³)	0.00642	
Número de Aletas	13	13
Espaçamento entre Aletas (m)	0.0038	0.008
Altura das Aletas (m)	0.03	0.02
Espessura das Aletas (m)	0.0016	0.0016
Área do Corte Transversal (m ²)	0.001327	0.0026
Diâmetro Hidráulico (m)	0.0151	0.025
Ventoinha Orion (OD4006 e OD9220)		
Área (m ²)	0.0016	0.008464
Velocidade (RPM)	8000	3575
Motor DC		
Resistência de Armadura (Ω)	1	1
Indutância de Armadura (μ H)	12	12
Constante de Força Contra-eletromotriz (V/RPM)	0.09375	0.149
Peltier (TEC1-12706)		
Área (m ²)	0.0016	
Seebeck (V/K)	0.02	
Condutividade Térmica (W/mK)	0.5	
Resistência (Ω)	0.85	

7 Resultados

Testou-se individualmente os blocos do sistema de caracterização projetado, mas não foi realizado um teste do sistema completo chaveando a bateria entre os componentes. Para o teste conectou-se uma bateria no canal 1 e uma fonte de bancada no canal 2. No canal 1 conectou-se somente o equipamento de espectroscopia e no canal 2 conectou-se a carga DC. Executou-se o código listado abaixo que realiza um ciclo de operação do sistema e registrou-se os dados lidos dos equipamentos com os módulos em Python implementados. Os dados coletados desse teste constam nas figuras 33, 34 e 35.

```
from dcl.dcl import DCL
from eis.eis import EIS
from mcu.mcu import MCU

def main():

    dcl_instance = DCL()
    eis_instance = EIS()
    mcu_instance = MCU()

    mode = "CURR"
    list_values = [0.6, 0.5, 0.2, 0.5, 0.01]
    channels = '1'
    list_time = [1.5, 1, 1, 1, 1]
    repetitions = 1
    sample_time = 0.01
    filename_dcl = "DCL_data.csv"
    filename_eis = "EIS_data.csv"

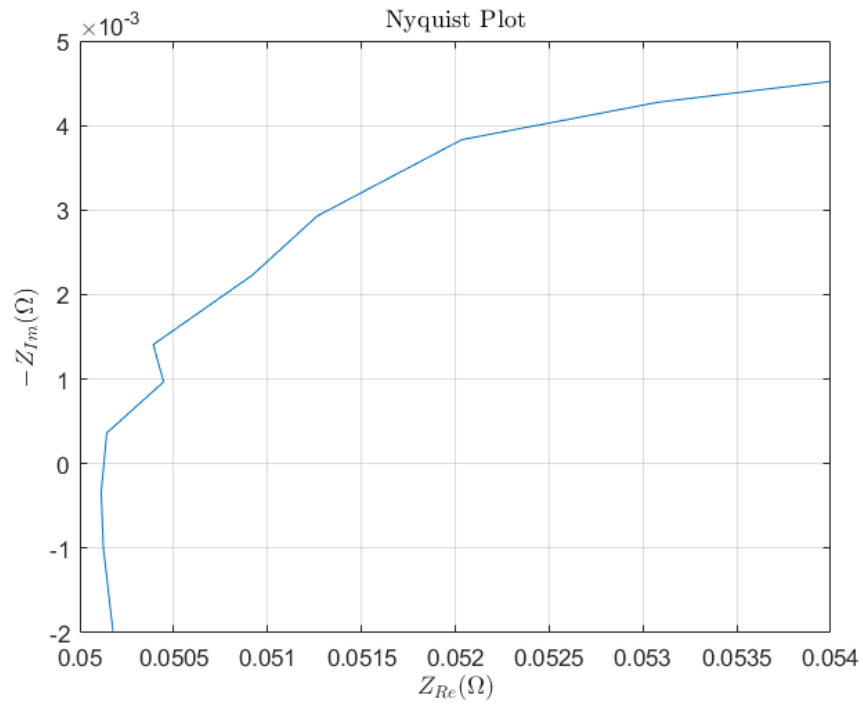
    mcu_instance.dcl_eis(False)
    df_eis = eis_instance.run(filename=filename_eis)
    mcu_instance.dcl_eis(True)
    df_dcl = dcl_instance.run(mode, list_values, channels=channels,
                             list_time=list_time, repetitions=
                                 repetitions,
                             sample_time=sample_time, filename=
                                 filename_dcl)

    print(df_dcl.head())

if __name__ == "__main__":
    main()
```

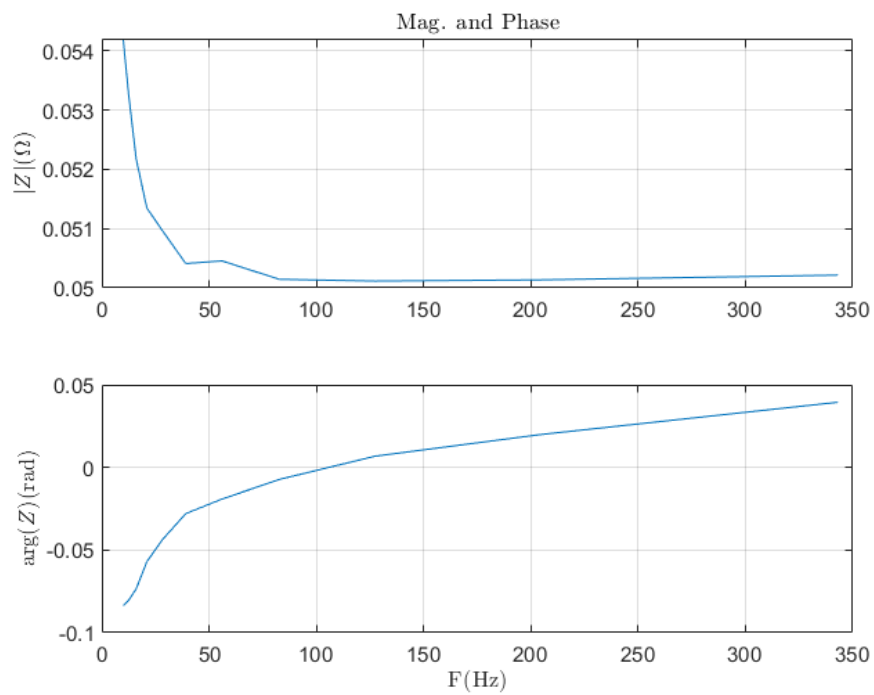
Listagem 7.1 – Código de teste da plataforma de caracterização.

Figura 33 – Diagrama de Nyquist do teste de espectroscopia.



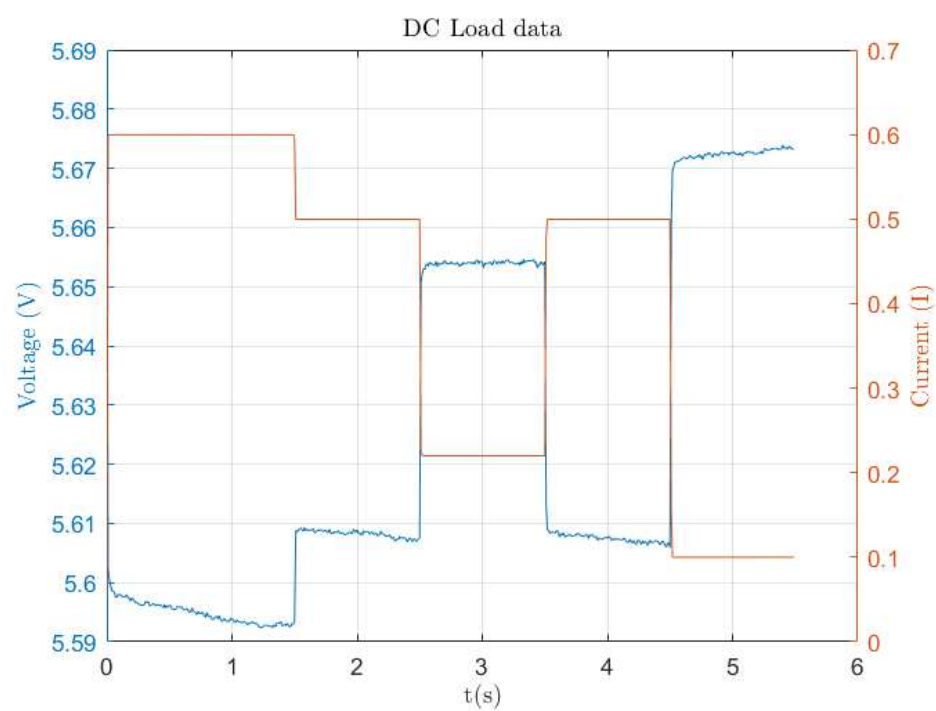
Fonte: Autoria própria.

Figura 34 – Magnitude e fase do teste de espectroscopia.



Fonte: Autoria própria.

Figura 35 – Dados da carga DC coletados no teste.



Fonte: Autoria própria.

8 Considerações Finais

Neste relatório foram descritas as principais atividades realizadas em um projeto de pesquisa e desenvolvimento como parte da disciplina de Estágio Supervisionado no Laboratório Embedded da UFCG. Com as ferramentas implementadas pode-se realizar ensaios intercalados de envelhecimento e caracterização automatizados, sem ser necessária a desconexão da bateria.

A câmara térmica, projetada e em processo de confecção permitirá a realização dos experimentos em uma bancada e tornará o uso do banho térmico dispensável.

A carga eletrônica DC possui inúmeras funções que ainda precisam ser exploradas mais a fundo, como por exemplo a execução de descargas de corrente pulsada programadas e os testes de bateria. Sua alta precisão e confiabilidade são indispensáveis para a caracterização da bateria e permitem a coleta de dados mais precisos durante a ensaios de envelhecimento.

O equipamento de EIS empregado no projeto promete leituras confiáveis na faixa de 1Hz à 200kHz. No entanto, nos testes realizados em laboratório, só foi possível obter dados menos ruidosos para frequências acima de 10Hz. Tendo em vista que é de interesse estimar a impedância de Warburg, faz-se necessário implementar uma solução que forneça dados confiáveis em faixas de sub-Hz.

A plataforma automática de ciclagem e caracterização ainda está em desenvolvimento como parte de um projeto do CNPq, e tem previsão de conclusão para Julho de 2024.

Com o conhecimento adquirido ao longo da graduação, foi possível analisar os problemas e propor soluções, bem como compreender temas mais complexos e que vão além do conteúdo apreendido em sala de aula, como modelagem de baterias e técnicas de caracterização de células eletroquímicas.

Referências

Analog Devices. *Electrochemical Impedance Spectroscopy (EIS) for Batteries*. [S.l.], 2022. Circuit Note CN-0510. Disponível em: <<https://www.analog.com/CN0510>>. Citado 7 vezes nas páginas 3, 6, 7, 10, 12, 14 e 15.

Analog Devices Inc. *AD5940 Examples*. 2024. <<https://github.com/analogdevicesinc/ad5940-examples>>. Accessed: 2024-06-03. Citado na página 16.

Bit Maker. *Módulo Relé 5V 10A 8 Canais com Optoacoplador*. 2024. <<https://www.bitmaker.com.br/modulo-rele-5v-10a-8-canais.html>>. Disponível para compra online, este módulo é ideal para projetos de automação, permitindo o controle de dispositivos via microcontroladores. Citado na página 34.

Clean Energy Institute. *Lithium-Ion Battery*. 2024. Clean Energy Institute, University of Washington. Disponível em: <<https://www.cei.washington.edu/research/energy-storage/lithium-ion-battery/>>. Citado na página 4.

IOT Robótica. *ESP32 Dev Kit*. [S.l.], 2024. Placa de desenvolvimento ESP32 NodeMcu para projetos de robótica e automação com WiFi, Bluetooth BLE 4.2, e suporte para programação via IDE Arduino. Disponível em: <<https://www.iot-robotica.com.br/produto/esp32-dev-kit.html>>. Citado na página 36.

IURILLI, P.; BRIVIO, C.; WOOD, V. On the use of electrochemical impedance spectroscopy to characterize and model the aging phenomena of lithium-ion batteries: a critical review. *Journal of Power Sources*, v. 505, p. 229860, 2021. ISSN 0378-7753. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0378775321003992>>. Citado na página 9.

Keysight Technologies. *DC Electronic Loads EL30000 Series User Guide*. [S.l.], 2024. User guide for the EL30000 series DC electronic loads. Disponível em: <<https://www.keysight.com>>. Citado 2 vezes nas páginas 22 e 23.

Keysight Technologies. *EL30000 Series DC Electronic Loads Programming Guide*. [S.l.], 2024. Programming guide for the EL30000 series DC electronic loads. Disponível em: <<https://www.keysight.com>>. Citado 3 vezes nas páginas 24, 25 e 30.

Keysight Technologies. *EL34243A 600W Bench Electronic Load, Dual Input 150V, 60A, 300W*. 2024. <<https://www.keysight.com/br/pt/product/EL34243A/dc-electronic-load-dual-input-2x-150v-60a-300w-lan-usb.html>>. Accessed: 2024-06-03. Citado 3 vezes nas páginas 3, 11 e 22.

KYUNGSOON, P. Impedance technique-based label-free electrochemical aptasensor for thrombin using single-walled carbon nanotubes-casted screen-printed carbon electrode. *Sensors*, v. 22, n. 7, 2022. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/22/7/2699>>. Citado na página 6.

MEDDINGS, N. et al. Application of electrochemical impedance spectroscopy to commercial li-ion cells: A review. *Journal of Power Sources*, v. 480, p. 228742, 2020.

ISSN 0378-7753. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0378775320310466>>. Citado na página 8.

Power Sonic. *Lithium Battery Configurations and Types of Lithium Cells*. 2024. Power Sonic. Disponível em: <<https://www.power-sonic.com/blog/lithium-battery-configurations/>>. Citado na página 4.

RAKHMATOV, D.; VRUDHULA, S. An analytical high-level battery model for use in energy management of portable electronic systems. p. 488–493, 2001. Citado na página 5.

TAVARES, A. d. H. B. M. *Analysis and Design of the Flight Control System for Battery Powered Mini Drones*. Tese (Doutorado) — Universidade Federal de Campina Grande, Campina Grande, Abril 2023. Citado na página 5.